**Report-HW1**

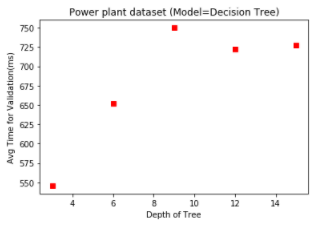
**(Abhishek Singhal)**

**Q1)**

1. The criteria for selecting a variable, as node in the tree, is minimizing the mean square error of the two splitting regions. We choose a variable and try all possible splits in the datum. Then we do this process for all variables. We select the variable(j) and split(s) s.t. the mean square error of the region to be divided, is minimum.

For tree of depth=1, it is optimal as we look at all the possible cases.

For trees of depth>1, we do recursive splitting. As naïve splitting is computationally expensive. No, it is not optimal as we have not searched all possible trees.



1. **Power Plant**

model chosen: max\_depth=9, criterion=’mae’

out of sample error=0.160256966417

full training error=0.118336262681

Out of Sample error is close to Full training error.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model (depth) | Error1 | Error2 | Error3 | Error4 | Error5 | Avg. Error |
| 3 | 0.21363307694986075 | 0.21213404111498263 | 0.21483569790940768 | 0.21206802787456452 | 0.2074905393728223 | 0.21203227664432758 |
| 6 | 0.17165978655988864 | 0.17703962125435538 | 0.1741445038327526 | 0.16561086898954705 | 0.17427250139372816 | 0.17254545640605437 |
| 9 | 0.15516345369080786 | 0.16125527909407672 | 0.16353107804878048 | 0.16053877177700346 | 0.16079624947735191 | **0.16025696641760409** |
| 12 | 0.16517684087743736 | 0.15946469616724745 | 0.16460551567944243 | 0.16203833275261331 | 0.15549862473867593 | 0.16135680204308328 |
| 15 | 0.16459260759052918 | 0.17032658989547039 | 0.16705502473867595 | 0.16769339268292685 | 0.17000531777003486 | 0.16793458653552742 |

1. **Indoor Localization**

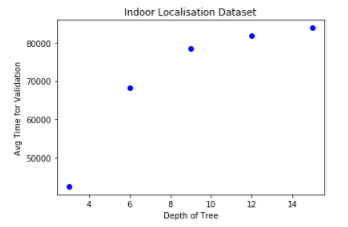
model chosen: max\_depth=15

out of sample error=9.0142990917955039

full training error=7.48820330784

Out of Sample error is not very close to Full training error.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model (depth)** | **Error1** | **Error2** | **Error3** | **Error4** | **Error5** | **Avg. Error** |
| **3** | 40.316933000376132 | 40.59329762738215 | 40.015660284926007 | 40.679415554176074 | 41.302495941183849 | 40.58156048160884 |
| **6** | 27.48125589568706 | 26.50964505692076 | 28.268997951718084 | 27.535988266804615 | 26.338258458490092 | 27.22682912592412 |
| **9** | 17.423034897881145 | 18.091903612274322 | 17.485970172686233 | 17.789486359794331 | 18.237538454602461 | 17.805586699447694 |
| **12** | 12.23399103435306 | 11.777981722291875 | 12.882466373651869 | 12.241183331640332 | 12.937726133182846 | 12.414669719023996 |
| **15** | 8.8434292245486468 | 9.0379293320586758 | 9.0100572337597189 | 9.0826730540506642 | 9.0974066145598194 | **9.0142990917955039** |



**Q2)**

1. N samples. Each chunk of size =M.

k=N/M is the number of subsets of training dataset.

We will train (N-M) samples for k times.

Total samples trained = (N-M)X(N/M)

Time taken ~ N2/M – N

Complexity ~ O(N2/M)

If M=5 , complexity is O(N2/5).

If M=N/2, then running time is linear. O(2N)

1. When we make M small, we actually increase k in k-fold cross validation. With this, the bias of estimated error will be small. Or we can say estimator can be fairly accurate.

**Q3.KNN**

1. **Power Plant**

model chosen n=3

out of sample error=0.22130739350784454

full training error= 0.361530806

Out of Sample error is close to Full training error.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model (n\_neighbors)** | **Error1** | **Error2** | **Error3** | **Error4** | **Error5** | **Avg. Error** |
| **3** | 0.5437207207520891 | 0.37095100255427843 | 0.19186524423285342 | 8.9421203028509396e-16 | 8.8916858516244428e-16 | **0.22130739350784454** |
| **5** | 0.53964015779944297 | 0.42937341086729358 | 0.33034079823502088 | 0.23170993457585784 | 0.11509728272017836 | 0.32923231683955867 |
| **10** | 0.58135280278551538 | 0.48551981532567046 | 0.4240396581049698 | 0.37354141827207804 | 0.31883918442028991 | 0.43665857578170469 |
| **20** | 0.60848649233983287 | 0.55284522499129218 | 0.5021451544356712 | 0.45413709683852982 | 0.42235784244704577 | 0.50799436221047434 |
| **25** | 0.58888568292479115 | 0.55646554952281435 | 0.51533930379006032 | 0.47546261017244379 | 0.44636510152731329 | 0.51650364958748463 |

1. **Indoor Localization**

model chosen n=3

out of sample error=1.60727811

full training error=2.2239872

Out of Sample error is not very close to Full training error.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model (n\_neighbors)** | **Error1** | **Error2** | **Error3** | **Error4** | **Error5** | **Avg. Error** |
| **3** | 3.4594528604563703 | 2.0966570466399213 | 1.3172279068795469 | 0.56479106005224922 | 0.59826170301449833 | **1.6072781154085174** |
| **5** | 3.8368866274322975 | 2.6133712489343037 | 2.0162536138092459 | 1.421127889768026 | 1.0011745483121846 | 2.1777627856512116 |
| **10** | 4.7261048981569713 | 3.3950221586822993 | 2.733493590441364 | 2.2961816310783743 | 1.9353148300747409 | 3.01722342168675 |
| **20** | 5.7976199270310937 | 4.4026279488622118 | 3.6568139681183669 | 3.1565055953714785 | 2.8143532027662208 | 3.9655841284298745 |
| **25** | 6.0613444560481442 | 4.814073318698596 | 3.982288090134583 | 3.4688552833918527 | 3.0905037186056132 | 4.283412973375758 |

**Q4. Linear Model**

1. The purpose of penalties in Ridge and Lasso regression is to control capacity. In several cases, all the attributes (p) of input data are not useful. The penalties makes coefficient βi small for **i**thattribute of **X.** The larger the penalty (α), the smaller the value of β.

In ridge regression, βs are made small but do not converge to zero. That is, no input attribute is ignored.

In lasso regression, β can be zero. This a selection of attributes occur.

1. **Power Plant**

Model chosen: alpha=0.01 (Lasso)

Out of sample error=0.18776822

Full training error=0.190764

Out of Sample error is close to Full training error.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Lasso (alpha)** | **Error1** | **Error2** | **Error3** | **Error4** | **Error5** | **Avg. Error** |
| **10^-6** | 0.18978352457554071 | 0.19398538069880425 | 0.19186026921584834 | 0.19080004536720838 | 0.19067876492417468 | 0.19142159695631528 |
| **10^-4** | 0.19175507937722855 | 0.18887131455626152 | 0.19005989567954601 | 0.19108683819736783 | 0.19067941628063609 | 0.190490508818208 |
| **10^-2** | 0.18443835440369816 | 0.18630927954994064 | 0.18757889274669534 | 0.18974968857416435 | 0.19076488768205507 | **0.18776822059131071** |
| **1** | 0.2666603571793485 | 0.26434001211203639 | 0.26405486766376685 | 0.26218930616776909 | 0.26245328777656507 | 0.26393956617989722 |
| **10** | 0.62329620576366584 | 0.62148947633336626 | 0.6236560027055742 | 0.62651651292788568 | 0.62768949695425558 | 0.62452953893694951 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ridge(alpha)** | **Error1** | **Error2** | **Error3** | **Error4** | **Error5** | **Avg. Error** |
| **10^-6** | 0.1919571695636817 | 0.19328834500600034 | 0.19317740671953548 | 0.19195673770276014 | 0.19067875833731485 | 0.19221168346585851 |
| **10^-4** | 0.18792490646272467 | 0.19103736119286774 | 0.19094632474772946 | 0.18989839829986058 | 0.19067875834024595 | 0.19009714980868567 |
| **10^-2** | 0.19414361825633891 | 0.19060084353701598 | 0.19089328768394551 | 0.19040849808191673 | 0.19067875863337849 | 0.19134500123851911 |
| **1** | 0.18650679327417996 | 0.18803605998555781 | 0.18933126988930352 | 0.19125137117950047 | 0.19067878794657855 | 0.18916085645502406 |
| **10** | 0.18752966366219898 | 0.18996838054764123 | 0.19006532963670314 | 0.19153979273873339 | 0.1906790544274605 | 0.18995644420254745 |

1. **Indoor Localization**

Model chosen: alpha=10 (Ridge)

Out of sample error= 18.970032206799651

Full training error=18.6044577636

Out of Sample error is very close to Full training error.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Lasso (alpha)** | **Error1** | **Error2** | **Error3** | **Error4** | **Error5** | **Avg. Error** |
| **10^-6** | 20.47708049290879 | 20.634091295324243 | 20.058269918698816 | 19.878756182890246 | 19.931362865996984 | 20.195912151163817 |
| **10^-4** | 20.151478357417044 | 19.455128957219078 | 20.460186593805226 | 20.777356272540423 | 20.170559352385183 | 20.202941906673392 |
| **10^-2** | 19.94083987632904 | 20.438487299950857 | 20.041168766019013 | 20.191730583255904 | 20.167033793553983 | 20.155852063821762 |
| **1** | 21.44192350345271 | 21.341933750531748 | 20.77250139021557 | 20.991333011626896 | 21.237534691471264 | 21.157045269459637 |
| **10** | 35.476969491427617 | 35.613805855557366 | 35.329753256371198 | 34.982710551830351 | 35.522383882598582 | 35.385124607557024 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ridge (alpha)** | **Error1** | **Error2** | **Error3** | **Error4** | **Error5** | **Avg. Error** |
| **10^-6** | 19.204266483693473 | 19.31476104522444 | 18.488004720819927 | 19.133594596324411 | 18.753334895918702 | 18.97879234839619 |
| **10^-4** | 19.006484640804544 | 18.824139184243098 | 19.038591711638318 | 18.980103339348965 | 19.029139946428252 | 18.97569176449263 |
| **10^-2** | 19.039378468528934 | 18.920775806994008 | 19.192728377638382 | 18.986285827973809 | 18.764728417721177 | 18.98077937977126 |
| **1** | 18.941915583117915 | 18.815454068655015 | 19.136983595819675 | 19.079018849427069 | 19.039240045015809 | 19.002522428407097 |
| **10** | 18.874853302804265 | 18.814607148918189 | 19.084361717063626 | 18.752438823573648 | 19.323900041638517 | **18.970032206799651** |