CS18M519 Programming Assignment 1 Report

Roll No: CS18M519
Name: Pawan Suresh Bathe

Q1.

After running python file from Code\q1\run.py, generated database will be stored in Dataset directory with name DS1-test.csv and DS1-train.csv

Q2.

Run *python file from Code\q2\run.py, it will build a linear model.*

Regression Coefficients of Learned Model:

[[-0.75981361 2.60216518 -0.50340592 2.26978113 1.11261818 -3.26152784

-1.59194991 -2.28940085 -0.42117328 -0.14369441 -0.63033349 3.22879494

1.68694538 -0.47397635 1.24645268 2.13699768 -2.08474517 0.72993358

-1.95441419 -0.57097383]]

Best Fit Accuracy Score: 0.843

Classification Report:

	precision	recall	f1-score	support
0.0	**0.86**	**0.82**	**0.84**	**602**
1.0	**0.83**	**0.86**	**0.85**	**598**

micro avg	0.84	0.84	0.84	1200
macro avg	0.84	0.84	0.84	1200
weighted avg	0.84	0.84	0.84	1200

Q3. Answer: Run *python file from Code\q3\run.py*

Upon experimentation it is found out that model perform better for K value in range of 30 - 40

Report For K=30:

Accuracy Score: 0.7483333333333333

Classification Report:

	precision	recall	f1-score	support
0.0	0.76	0.73	0.74	602
1.0	0.74	0.76	0.75	598
micro avg	0.75	0.75	0.75	1200
macro avg	0.75	0.75	0.75	1200
weighted avg	0.75	0.75	0.75	1200

Report For K=31:

Accuracy Score: 0.755

Classification Report:

	precision	recall	f1-score	support
0.0	0.76	0.75	0.75	602
1.0	0.75	0.76	0.76	598

micro avg	0.76	0.76	0.76	1200
macro avg	0.76	0.76	0.75	1200
weighted avg	0.76	0.76	0.75	1200

Report For K=32:

Accuracy Score: 0.7575

Classification Report:

	precision	recall	f1-score	support
0.0	0.76	0.75	0.76	602
1.0	0.75	0.77	0.76	598
micro avg	0.76	0.76	0.76	1200
macro avg	0.76	0.76	0.76	1200
weighted avg	0.76	0.76	0.76	1200

Report For K=33:

Accuracy Score: 0.7625

Classification Report:

	precision	recall	f1-score	support
0.0	0.77	0.75	0.76	602
1.0	0.75	0.78	0.77	598
micro avg	0.76	0.76	0.76	1200
macro avg	0.76	0.76	0.76	1200
weighted avg	0.76	0.76	0.76	1200

Report For K=34:

Accuracy Score: 0.7558333333333334

Classification Report:

	precision	recall	f1-score	support

0.0	**0.77**	**0.74**	**0.75**	**602**
1.0	**0.75**	**0.77**	**0.76**	**598**
micro avg	**0.76**	**0.76**	**0.76**	**1200**
macro avg	**0.76**	**0.76**	**0.76**	**1200**
weighted avg	**0.76**	**0.76**	**0.76**	**1200**

So we report value for K=33 which turned out to be best:

• Do you do better than regression on indicator variables or worse?

Ans. KNN performs worse than regression on indicator variables

• Are there particular values of k which perform better?

Ans: For K =33 model performs better

• Report the best fit accuracy, precision, recall and f-measure achieved by this clas- sifier.

Report For Best K=33:

Accuracy Score: 0.7625

Classification Report:

	precision	**recall**	**f1-score**	**support**
0.0	**0.77**	**0.75**	**0.76**	**602**
1.0	**0.75**	**0.78**	**0.77**	**598**
micro avg	**0.76**	**0.76**	**0.76**	**1200**
macro avg	**0.76**	**0.76**	**0.76**	**1200**
weighted avg	**0.76**	**0.76**	**0.76**	**1200**

Q4. Answer: Run *python file from Code\q4\run.py* it will impute missing values by using mean value of missing feature and by interpolating the missing feature value

• Use the sample mean of the missing attribute. Is this is a good choice?

Ans.: dataset after imputing mean value is stored in CandC-Imputed_mean.csv, for this use case means seems to be working well for imputation

• What else might you use? If you have a better method, describe it, and you may use it for filling in the missing data. Turn in the completed data set.

Ans.: Other that mean, we can use several other metrics like max, min, most_frequent occurred values or interpolation on missing attributes.

Dataset by imputing missing values by interpolation is saved in CandC-Imputed_interpolate.csv.

Mean seems to worked well for our case ,we will use mean imputed dataset for further experiments.

Q5 Answer:

Run *python file from Code\q5\run.py*.

RSS for 5 splits is [8.268062773830424, 7.534484194801976, 7.4513014008835015, 733.1872661641842, 8.11691754323708]

Lowest RSS for best fit is 7.4513014008835015 for split number 3

Coefficients learned for this best fit are :

[0.03174554 0.03936066 0.16006474 -0.05855977 -0.04163412

0.05517187 0.09430254 -0.19916523 -0.14227983 -0.02385112

-0.01710755 0.03929021 -0.18536458 -0.20096267 0.06564561

-0.15425707 0.16226102 0.0270615 -0.07764927 0.34188228

0.0780521 -0.29075141 -0.02660168 -0.03985609 0.02488996

0.04398538 0.03428011 -0.03861777 -0.11952524 -0.1453211

0.06925507 0.05376095 0.00105432 0.26788053 -0.04879769

-0.00716152 0.06538726 0.01532664 0.13201723 0.19193642

-0.15857925 0.06190561 -0.03645793 0.08428583 -0.39322681

-0.00810806 -0.00497482 0.08626803 -0.19797509 -0.15453402

 0.14963232 -0.15087809 0.00692553 0.06843202 -0.15219644

 0.08040356 -0.03452837 -0.22465511 0.48452884 -0.21849523

-0.05182821 -0.16532767 -0.13912939 0.02637786 0.3934253

-0.05764484 -0.25345898 -0.7606255   0.17777814 0.04737364

 0.00996061 0.17639708 -0.04324872 0.63339911 0.06145501

-0.06565555 -0.01256342 0.00283289 -0.01288589 -0.1444498

-0.01087169 0.07896048 -0.25578968 0.02537195 -0.11185662

 0.39964138 0.02444557 -0.01041736 -0.09353009 0.11393876

 0.14047137 0.13593265 0.01196304 -0.01899999 0.02186998

-0.01116575 -0.80667731 -17.79530999 -0.60059362 0.18585816

-0.03648765 -0.01619634 0.09055507 17.78924375 -0.0584019

 0.09275461 -0.04308214 -0.03132011 0.02558615 0.22288321

-0.25887937 0.00305885 -0.01625539 0.06000402 -0.00069573

-0.01964658 -0.08038527 0.56731608 0.00080623 0.03652967

 0.02590237 -0.10456048]

Average RSS over 5 splits is 152.912

1. Make 5 different 80-20 splits in the data and name them as CandC−train〈num〉.csv and CandC − test〈num〉.csv.

Ans: CandC-train1.csv to CandC-train5.csv and CandC-test1.csv to CandC-test5.csv are stored in Dataset directory.

2. For all 5 datasets that you have generated, learn a regression model using 80% and test it using 20%.

Ans. After learning model for 5 datasets RSS for 5 splits is:

[8.268062773830424,7.534484194801976,7.451301400835015, 733.1872661641842, 8.11691754323708]

3. Report the average RSS over these 5 different runs.

Ans: Average RSS over 5 splits is 152.912

Regularized Linear Regression

6. Use Ridge regression on the CandC data. Repeat the experiment for various values of λ

• Report the residual error for each value, on test data, averaged over 5 different 80-20 splits, along with the coefficients learnt.

Ans:

Lambda = 1:

RSS for 5 splits is [7.9859557870784315, 7.507625675700385,7.542530526887831, 6.355396649976413, 8.237574664809838]

Lowest RSS for best fit is 6.355396649976413 for split number 4

Coefficients learned for this best fit are :

[-0.01031088 0.02866945 0.16813252 -0.05050414 -0.0361528 0.06360787

0.03888323 -0.18017719 -0.02966441 0.04841939 -0.02392568 0.04502607

0.05302073 -0.07441275 0.03262481 -0.15512609 0.09188954 0.01143691

-0.10002599 0.03040527 -0.00031281 -0.14608683 -0.02605055 -0.03749227

0.01714786 0.03958186 0.04625224 -0.01585585 -0.16692092 -0.04994531

0.02457813 0.03210746 0.0093492   0.10863559 -0.03514721 -0.00235532

0.04545341 0.05452774 0.16679495 0.14075638 -0.08977553 -0.01339336

0.00861682 -0.07174741 -0.17987511 -0.04358276 -0.01256115 0.03628479

-0.15204957 -0.06387114 0.11630738 -0.09538997 0.00412945 -0.01273717

-0.02293605 0.03272586 -0.00979003 0.02329954 0.0571007   0.00199507

0.02221566 -0.14414502 -0.05985484 -0.03507179 0.18403363 -0.05112341

-0.02504741 -0.10254332 0.12205885 0.09346037 0.0284101   0.10537689

-0.06558513 0.01661519 0.06905405 -0.06853644 -0.01443857 0.02472366

-0.00054531 -0.08332663 0.02421831 0.02260401 -0.16380941 0.01175959

0.00602897 0.16381352 0.07965853 -0.02748058 -0.09825639 0.12607846

0.14945509 0.08937181 -0.00162733 0.00267687 0.02478992 0.0155686

-0.01663062 0.02609099 -0.01488108 0.0679132 -0.06973288 0.01694927

0.09152783 0.02602893 -0.03685816 -0.05311909 -0.02234862 0.04330932

0.07208312 -0.02472105 -0.08461137 0.00016622 -0.00774539 0.07255466

-0.00382841 -0.03205673 0.03448152 0.04735823 0.03116802 0.04273475

0.02856135 -0.04841322]

Average RSS over 5 splits is 7.526

Lambda = 2:

RSS for 5 splits is [7.959230416352504, 7.51767933812917, 7.6029487025514335, 6.3903641364883 1, 8.296301546047282]

Lowest RSS for best fit is 6.39036413648831 for split number 4

Coefficients learned for this best fit are :

[-0.00957509 0.03029915 0.155998 -0.06152181 -0.03854662 0.05028303
0.02041865 -0.12479296 -0.02533075 0.04598628 -0.0162032 0.04444886
0.04398798 -0.05647284 0.02798031 -0.12955061 0.06693149 0.01376048
-0.09110251 0.01801567 -0.01269764 -0.09338811 -0.02415241 -0.0363402
0.01782614 0.03860538 0.04363284 -0.02040234 -0.13817496 -0.04396913
0.02521913 0.0176737 0.00167906 0.06928086 -0.03013686 0.00222152
0.03687779 0.03338605 0.13353891 0.10400089 -0.05890296 0.00551673
0.01470637 -0.07493722 -0.14712627 -0.05226033 -0.01714751 0.02106927
-0.12425021 -0.04393422 0.12183574 -0.06590257 0.00232516 -0.01309532
-0.01701444 0.02436677 -0.0033822 0.02000813 0.0430814 0.01989584
0.02409643 -0.11132347 -0.03884394 -0.02712418 0.10964929 -0.03102186
-0.00666591 -0.06674322 0.09560564 0.07652305 0.02272621 0.09305575
-0.06823498 0.00514665 0.06811637 -0.06201291 -0.01191997 0.02551837
0.00041564 -0.05224489 0.0099383 0.0148679 -0.13465639 0.0143715
0.02659369 0.1135738 0.0735983 -0.018802 -0.09400005 0.10408613
0.14032609 0.06748499 -0.00523425 0.00510928 0.02831483 0.0137691
-0.01713928 0.01921787 -0.00030735 0.04285509 -0.0460907 0.02318738
0.07412977 0.01918579 -0.03440332 -0.04440283 -0.02194337 0.0318159
0.06341498 -0.0129111 -0.05130525 0.00171727 -0.00759981 0.06371672
-0.00360471 -0.02703603 0.032634 0.01780104 0.03532121 0.04176895

0.02796943 -0.01896092]

Average RSS over 5 splits is 7.553

Lambda = 3:

RSS for 5 splits is [7.94948226867866, 7.531911165964843, 7.646363725502018, 6.418506436455072, 8.337775666266914]

Lowest RSS for best fit is 6.418506436455072 for split number 4

Coefficients learned for this best fit are :

[-0.00757315 0.02820598 0.14900504 -0.06771036 -0.03877977 0.04171879

0.01151573 -0.09730306 -0.02264198 0.0412625 -0.01175905 0.04400335

0.03562084 -0.04681122 0.02463238 -0.11333725 0.05424537 0.01493982

-0.08338546 0.01230096 -0.0136173 -0.06731615 -0.02306125 -0.03541444

0.01803844 0.0375798 0.04149778 -0.01903262 -0.11730816 -0.03909743

0.02446982 0.00872717 -0.00181582 0.04879634 -0.02704964 0.00369709

0.03155108 0.02213676 0.11544822 0.08464295 -0.03944726 0.01473068

0.01470631 -0.07374128 -0.13044272 -0.05621992 -0.02052287 0.01246307

-0.10647003 -0.03278743 0.12364312 -0.04919222 0.00103154 -0.01282796

-0.01403443 0.02033464 -0.00017015 0.01887292 0.03749658 0.02496882

0.02420147 -0.09176906 -0.0264957 -0.02060761 0.07880452 -0.02301115

-0.00076785 -0.05080856 0.0807343 0.06664732 0.01914088 0.08399687

-0.06910609 0.00217851 0.06697488 -0.05705628 -0.00997333 0.0257415

0.00101488 -0.03875127 0.00596976 0.01181021 -0.11446233 0.0141773

0.03202865 0.0891463 0.06870493 -0.01319201 -0.09007381 0.09005091

0.13123872 0.05557616 -0.00740561 0.00512903 0.02965297 0.01306589

-0.01660285 0.01570197 0.00413288 0.03287696 -0.03444099 0.02545114

0.06495301 0.01568006 -0.03284851 -0.03845724 -0.01937083 0.0265391

0.0581303 -0.00815463 -0.03754091 0.00240102 -0.0076057 0.05718399

-0.00324359 -0.023478 0.03027487 0.00886892 0.03588131 0.04115123

0.02829941 -0.00746065]

Average RSS over 5 splits is 7.577

Lambda = 4:

RSS for 5 splits is [7.945540312477627, 7.545247978200661, 7.679092249203672, 6.441316673584081, 8.370161633285733]

Lowest RSS for best fit is 6.441316673584081 for split number 4

Coefficients learned for this best fit are :

[-0.00557609 0.02570443 0.1442383 -0.07158948 -0.03821728 0.03573381

0.00639681 -0.08074148 -0.02085681 0.03710209 -0.00851922 0.04366387

0.02952786 -0.04076957 0.02200171 -0.10187885 0.04619565 0.01579018

-0.07675608 0.00887344 -0.01254162 -0.05128091 -0.02230462 -0.03455102

0.01811852 0.03663055 0.03968051 -0.0166032 -0.10139438 -0.03515269

0.02364481 0.00303072 -0.00346527 0.03635406 -0.02493317 0.00393018

0.02792256 0.01516956 0.10387931 0.07240775 -0.02621988 0.02045964

0.01387268 -0.07218633 -0.11983286 -0.05827931 -0.02325697 0.00709219

-0.0939418 -0.0252505 0.12382681 -0.03829941 0.00004053 -0.01233264

-0.0121203 0.01783091 0.00187956 0.01827726 0.03416638 0.02667482

0.02384946 -0.07847318 -0.01821122 -0.01558491 0.06167767 -0.01884612

0.00185745 -0.04158598 0.07123861 0.0598494 0.01657018 0.07722527

-0.06911043 0.00102663 0.06589267 -0.0530094 -0.00823542 0.02584156

0.00146931 -0.03104095 0.00443081 0.01050943 -0.09947495 0.01352184

0.03299394 0.0742754 0.06480205 -0.00918856 -0.08652546 0.08014652

0.12304375 0.04783845 -0.00890229 0.00466171 0.03016175 0.0126456

-0.01551626 0.01358286 0.005674 0.02739069 -0.02711566 0.02628478

0.05891112 0.01356606 -0.03161403 -0.03415979 -0.01681847 0.02318703

0.05426446 -0.00543035 -0.02969095 0.00278435 -0.00758851 0.05210757

-0.00282861 -0.02078942 0.02819441 0.00517225 0.0352836 0.04065921

0.02881152 -0.00158753]

Average RSS over 5 splits is 7.596

• Which value of λ gives the best fit?

Ans: By experimentation it is observed that model with lambda value of 1 gives best fit. Use this value to retrieve top features and train model on small set of features:

Lambda = 1:

RSS for 5 splits is [7.9859557870784315, 7.507625675700385, 7.542530526887831, 6.355396649976413, 8.237574664809838]

Lowest RSS for best fit is 6.355396649976413 for split number 4

Coefficients learned for this best fit are :

[-0.01031088 0.02866945 0.16813252 -0.05050414 -0.0361528   0.06360787
0.03888323 -0.18017719 -0.02966441 0.04841939 -0.02392568 0.04502607
0.05302073 -0.07441275 0.03262481 -0.15512609 0.09188954 0.01143691
-0.10002599 0.03040527 -0.00031281 -0.14608683 -0.02605055 -0.03749227
0.01714786 0.03958186 0.04625224 -0.01585585 -0.16692092 -0.04994531
0.02457813 0.03210746 0.0093492   0.10863559 -0.03514721 -0.00235532
0.04545341 0.05452774 0.16679495 0.14075638 -0.08977553 -0.01339336
0.00861682 -0.07174741 -0.17987511 -0.04358276 -0.01256115 0.03628479
-0.15204957 -0.06387114 0.11630738 -0.09538997 0.00412945 -0.01273717
-0.02293605 0.03272586 -0.00979003 0.02329954 0.0571007   0.00199507
0.02221566 -0.14414502 -0.05985484 -0.03507179 0.18403363 -0.05112341
-0.02504741 -0.10254332 0.12205885 0.09346037 0.0284101   0.10537689
-0.06558513 0.01661519 0.06905405 -0.06853644 -0.01443857 0.02472366
-0.00054531 -0.08332663 0.02421831 0.02260401 -0.16380941 0.01175959
0.00602897 0.16381352 0.07965853 -0.02748058 -0.09825639 0.12607846
0.14945509 0.08937181 -0.00162733 0.00267687 0.02478992 0.0155686
-0.01663062 0.02609099 -0.01488108 0.0679132 -0.06973288 0.01694927
0.09152783 0.02602893 -0.03685816 -0.05311909 -0.02234862 0.04330932
0.07208312 -0.02472105 -0.08461137 0.00016622 -0.00774539 0.07255466
-0.00382841 -0.03205673 0.03448152 0.04735823 0.03116802 0.04273475
0.02856135 -0.04841322]

Average RSS over 5 splits is 7.526

• Is it possible to use the information you obtained during this experiment for feature selection? If so, what is the best fit you achieve with a reduced set of features?

Ans: We can retrieve the important features using most important attributes.

Features retrieved using regularized linear regression

{'MedRent', 'agePct12t29', 'racepctblack', 'pctWInvInc', 'PctPopUnderPov', 'PctPersOwnOccup', 'HousVacant', 'MedOwnCostPctIncNoMtg', 'pctWRetire', 'PctVacantBoarded', 'NumStreet', 'NumImmig', 'PctIlleg', 'PctPersDenseHous',

'PctNotSpeakEnglWell', 'pctWSocSec', 'RentLowQ', 'PctForeignBorn', 'PctWorkMom', 'whitePerCap', 'PctEmploy', 'PctKids2Par', 'PersPerOwnOccHous', 'MalePctDivorce', 'MalePctNevMarr', 'pctWWage', 'OwnOccLowQuart', 'PersPerOccupHous'}

RSS for 5 splits is [7.845937685699692, 7.497321637111819, 7.637080688016985, 6.2982447437881275, 8.026803482941832]

Lowest RSS for best fit is 6.2982447437881275 for split number 4 with reduced features.

Coefficients learned for this best fit are [0.31416903 -0.19806287 0.17733471 -0.11476448 -0.12390781 -0.03685232

0.16615888 -0.11792907 -0.09484774 0.05066868 0.18346203 -0.13582485

0.12655467 0.23211326 -0.18593564 0.05975994 -0.22195123 0.16760432

-0.13356546 -0.03408511 0.1018959 -0.27869343 -0.04685869 0.12868796

0.19428268 -0.11834867 -0.10102214 0.06695836]

Equation of fitted line is :

ViolentCrimesPerPop = 0.314 * population + -0.279 * whitePerCap + 0.232 * pctWWage + -0.222 * pctWSocSec + -0.198 * householdsize + 0.194 * AsianPerCap + -0.186 * pctWFarmSelf + 0.183 * numbUrban + 0.177 * racepctblack + 0.168 * pctWPubAsst + 0.166 * agePct12t21 + -0.136 * pctUrban + -0.134 * pctWRetire + 0.129 * indianPerCap + 0.127 * medIncome + -0.124 * racePctAsian + -0.118 * OtherPerCap + -0.118 * agePct12t29 + -0.115 * racePctWhite + 0.102 * perCapInc + -0.101 * HispPerCap + -0.095 * agePct16t24 + 0.067 * NumUnderPov + 0.06 * pctWInvInc + 0.051 * agePct65up + -0.047 * blackPerCap + -0.037 * racePctHisp + -0.034 * medFamInc

Average RSS over 5 splits for model with reduced features is 7.461

Q7 Answer:

Classification Report for Logistic Regression :

	precision	recall	f1-score	support
-1.0	0.90	0.95	0.92	19
1.0	0.95	0.90	0.93	21
micro avg	0.93	0.93	0.93	40
macro avg	0.93	0.93	0.92	40
weighted avg	0.93	0.93	0.93	40

Classification Report for L1 Regularized Logistic Regression :

	precision	recall	f1-score	support
-1.0	0.90	0.95	0.92	19
1.0	0.95	0.90	0.93	21
micro avg	0.93	0.93	0.93	40
macro avg	0.93	0.93	0.92	40
weighted avg	0.93	0.93	0.93	40