

Filterbasedpredictors

March 17, 2023

```
[1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.model_selection import train_test_split
from sklearn.feature_selection import mutual_info_regression, \
    mutual_info_classif
```

```
[103]: #read dataset
data = pd.read_csv("/content/drive/MyDrive/CIND 820 Capstone Project/
    merged_completedata.csv")
data.shape
```

[103]: (78032, 28)

```
[104]: # filter dataframe to 2019 records
data = data[data['Year'] >= 2019]
```

```
[105]: data.head()
```

```
[105]:
```

	RecordID	X	Y	FID	BusinessID	\
46689	46690	-79.665386	43.684736	1	7	
46690	46691	-79.642760	43.593515	2	4246	
46691	46692	-79.667311	43.682752	3	10	
46692	46693	-79.629235	43.698932	4	4247	
46693	46694	-79.629235	43.698932	5	4250	

	Name	Address	StreetNo	\
46689	Peel Car & Truck Rentals	7050 Bramalea Rd	7050	
46690	Real Fruit Bubble Tea	100 City Centre Dr	100	
46691	Unifor 2002	7015 Tranmere Dr	7015	
46692	Laura with Plus and Petites	100 City Centre Dr	100	
46693	Footlocker	100 City Centre Dr	100	

	StreetName	BldgNo	...	Fax	TollFree	EEmail	WebAddress	EmplRange	\
46689	Bramalea Rd	Yes	...	Yes	Yes	Yes	Yes	1	

46690	City Centre Dr	No	...	No	No	No	Yes	2
46691	Tranmere Dr	No	...	Yes	Yes	Yes	Yes	3
46692	City Centre Dr	No	...	No	No	No	Yes	2
46693	City Centre Dr	No	...	No	No	No	No	4

	CENT_X	CENT_Y	Year	isnew	Closed
46689	607567.2334	4.837723e+06	2019	No	No
46690	609556.5032	4.827621e+06	2019	Yes	No
46691	607415.6044	4.837500e+06	2019	No	No
46692	610454.8654	4.839347e+06	2019	Yes	No
46693	610454.8654	4.839347e+06	2019	Yes	No

[5 rows x 28 columns]

```
[106]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 31343 entries, 46689 to 78031
Data columns (total 28 columns):
#   Column          Non-Null Count  Dtype
---  -
0   RecordID        31343 non-null  int64
1   X                31343 non-null  float64
2   Y                31343 non-null  float64
3   FID              31343 non-null  int64
4   BusinessID      31343 non-null  int64
5   Name             31343 non-null  object
6   Address          31343 non-null  object
7   StreetNo         31343 non-null  int64
8   StreetName       31343 non-null  object
9   BldgNo           31343 non-null  object
10  UnitNo           31343 non-null  object
11  PostalCode       31343 non-null  object
12  Location         31343 non-null  object
13  Ward             31343 non-null  int64
14  NAICSCode        31343 non-null  int64
15  NAICSCat         31343 non-null  object
16  NAICSDescr       31343 non-null  object
17  Phone            31343 non-null  object
18  Fax              31343 non-null  object
19  TollFree         31343 non-null  object
20  EMail            31343 non-null  object
21  WebAddress       31343 non-null  object
22  EmplRange        31343 non-null  int64
23  CENT_X           31343 non-null  float64
24  CENT_Y           31343 non-null  float64
25  Year             31343 non-null  int64
```

```

26  isnew          31343 non-null  object
27  Closed         31343 non-null  object
dtypes: float64(4), int64(8), object(16)
memory usage: 6.9+ MB

```

```

[107]: #NAICSCode back to object as it is nominal not ordinal
data['NAICSCode'] = data['NAICSCode'].astype(str)

```

```

[108]: #drop unique fields
data.drop(['RecordID', 'FID', 'BusinessID', 'Name', 'Address',
↪ 'StreetNo', 'StreetName', 'NAICSDescr', 'Year'], axis=1, inplace=True)

```

```

[109]: data.info()

```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 31343 entries, 46689 to 78031
Data columns (total 19 columns):
#   Column          Non-Null Count  Dtype
---  -
0   X                31343 non-null  float64
1   Y                31343 non-null  float64
2   BldgNo           31343 non-null  object
3   UnitNo           31343 non-null  object
4   PostalCode       31343 non-null  object
5   Location         31343 non-null  object
6   Ward             31343 non-null  int64
7   NAICSCode        31343 non-null  object
8   NAICSCat         31343 non-null  object
9   Phone           31343 non-null  object
10  Fax              31343 non-null  object
11  TollFree         31343 non-null  object
12  EMail            31343 non-null  object
13  WebAddress       31343 non-null  object
14  EmplRange        31343 non-null  int64
15  CENT_X           31343 non-null  float64
16  CENT_Y           31343 non-null  float64
17  isnew            31343 non-null  object
18  Closed           31343 non-null  object
dtypes: float64(4), int64(2), object(13)
memory usage: 4.8+ MB

```

```

[110]: #describe categorical data
data.describe(include='O')
#There is none if I get an error

```

```

[110]:      BldgNo UnitNo PostalCode      Location NAICSCode  NAICSCat  \
count    31343   31343      31343      31343      31343      31343

```

unique	2	2	37	56	24	19
top	No	Yes	L4W Northeast EA (West)	81	Retail Trade	
freq	29769	20869	5084	8325	3576	4377

	Phone	Fax	TollFree	EMail	WebAddress	isnew	Closed
count	31343	31343	31343	31343	31343	31343	31343
unique	2	2	2	2	2	2	2
top	Yes	Yes	No	Yes	Yes	No	No
freq	30965	19310	26636	19963	23174	28576	28629

```
[111]: #if there is categorical data then factorize it
data['WebAddress'] = pd.factorize(data['WebAddress'])[0]
data['BldgNo'] = pd.factorize(data['BldgNo'])[0]
data['Fax'] = pd.factorize(data['Fax'])[0]
data['TollFree'] = pd.factorize(data['TollFree'])[0]
data['UnitNo'] = pd.factorize(data['UnitNo'])[0]
data['isnew'] = pd.factorize(data['isnew'])[0]
data['Closed'] = pd.factorize(data['Closed'])[0]
data['NAICSCode'] = pd.factorize(data['NAICSCode'])[0]
data['NAICSCat'] = pd.factorize(data['NAICSCat'])[0]
data['Location'] = pd.factorize(data['Location'])[0]
data['Phone'] = pd.factorize(data['Phone'])[0]
data['EMail'] = pd.factorize(data['EMail'])[0]
data['PostalCode'] = pd.factorize(data['PostalCode'])[0]
```

```
[112]: #describe categorical data
data.describe(include='O')
#There is none if I get an error
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-112-0c5d7d2ffda0> in <module>
      1 #describe categorical data
----> 2 data.describe(include='O')
      3 #There is none if I get an error

/usr/local/lib/python3.9/dist-packages/pandas/core/generic.py in describe(self,
↳ percentiles, include, exclude, datetime_is_numeric)
   10230         max          NaN          3.0
   10231         """
> 10232         return describe_ndframe(
   10233             obj=self,
   10234             include=include,

/usr/local/lib/python3.9/dist-packages/pandas/core/describe.py in
↳ describe_ndframe(obj, include, exclude, datetime_is_numeric, percentiles)
```

```

92         )
93
---> 94     result = describer.describe(percentiles=percentiles)
95     return cast(NDFrameT, result)
96
/usr/local/lib/python3.9/dist-packages/pandas/core/describe.py in describe(self,
↳ percentiles)
175
176     col_names = reorder_columns(ldesc)
--> 177     d = concat(
178         [x.reindex(col_names, copy=False) for x in ldesc],
179         axis=1,
180     )
181
/usr/local/lib/python3.9/dist-packages/pandas/util/_decorators.py in
↳ wrapper(*args, **kwargs)
309         stacklevel=stacklevel,
310     )
--> 311     return func(*args, **kwargs)
312
313     return wrapper
314
/usr/local/lib/python3.9/dist-packages/pandas/core/reshape/concat.py in
↳ concat(objs, axis, join, ignore_index, keys, levels, names, verify_integrity,
↳ sort, copy)
345     ValueError: Indexes have overlapping values: ['a']
346     """
--> 347     op = _Concatenator(
348         objs,
349         axis=axis,
350         join=join,
351         ignore_index=ignore_index,
352         keys=keys,
353         levels=levels,
354         names=names,
355         verify_integrity=verify_integrity,
356         copy=copy,
357         sort=sort,
358     )
359
/usr/local/lib/python3.9/dist-packages/pandas/core/reshape/concat.py in
↳ __init__(self, objs, axis, join, keys, levels, names, ignore_index,
↳ verify_integrity, copy, sort)
402
403     if len(objs) == 0:
--> 404         raise ValueError("No objects to concatenate")
405
406     if keys is None:
407         keys = [None] * len(objs)
408
409     if levels is None:
410         levels = [None] * len(objs)
411
412     if names is None:
413         names = [None] * len(objs)
414
415     if ignore_index is None:
416         ignore_index = False
417
418     if copy is None:
419         copy = False
420
421     if sort is None:
422         sort = False
423
424     self.objs = objs
425     self.axis = axis
426     self.join = join
427     self.ignore_index = ignore_index
428     self.keys = keys
429     self.levels = levels
430     self.names = names
431     self.verify_integrity = verify_integrity
432     self.copy = copy
433     self.sort = sort
434
435     self._validate()
436
437     self._concat()
438
439     self._finalize()
440
441     self._sort()
442
443     self._copy()
444
445     self._verify_integrity()
446
447     self._finalize()
448
449     self._sort()
450
451     self._copy()
452
453     self._verify_integrity()
454
455     self._finalize()
456
457     self._sort()
458
459     self._copy()
460
461     self._verify_integrity()
462
463     self._finalize()
464
465     self._sort()
466
467     self._copy()
468
469     self._verify_integrity()
470
471     self._finalize()
472
473     self._sort()
474
475     self._copy()
476
477     self._verify_integrity()
478
479     self._finalize()
480
481     self._sort()
482
483     self._copy()
484
485     self._verify_integrity()
486
487     self._finalize()
488
489     self._sort()
490
491     self._copy()
492
493     self._verify_integrity()
494
495     self._finalize()
496
497     self._sort()
498
499     self._copy()
500
501     self._verify_integrity()
502
503     self._finalize()
504
505     self._sort()
506
507     self._copy()
508
509     self._verify_integrity()
510
511     self._finalize()
512
513     self._sort()
514
515     self._copy()
516
517     self._verify_integrity()
518
519     self._finalize()
520
521     self._sort()
522
523     self._copy()
524
525     self._verify_integrity()
526
527     self._finalize()
528
529     self._sort()
530
531     self._copy()
532
533     self._verify_integrity()
534
535     self._finalize()
536
537     self._sort()
538
539     self._copy()
540
541     self._verify_integrity()
542
543     self._finalize()
544
545     self._sort()
546
547     self._copy()
548
549     self._verify_integrity()
550
551     self._finalize()
552
553     self._sort()
554
555     self._copy()
556
557     self._verify_integrity()
558
559     self._finalize()
560
561     self._sort()
562
563     self._copy()
564
565     self._verify_integrity()
566
567     self._finalize()
568
569     self._sort()
570
571     self._copy()
572
573     self._verify_integrity()
574
575     self._finalize()
576
577     self._sort()
578
579     self._copy()
580
581     self._verify_integrity()
582
583     self._finalize()
584
585     self._sort()
586
587     self._copy()
588
589     self._verify_integrity()
590
591     self._finalize()
592
593     self._sort()
594
595     self._copy()
596
597     self._verify_integrity()
598
599     self._finalize()
600
601     self._sort()
602
603     self._copy()
604
605     self._verify_integrity()
606
607     self._finalize()
608
609     self._sort()
610
611     self._copy()
612
613     self._verify_integrity()
614
615     self._finalize()
616
617     self._sort()
618
619     self._copy()
620
621     self._verify_integrity()
622
623     self._finalize()
624
625     self._sort()
626
627     self._copy()
628
629     self._verify_integrity()
630
631     self._finalize()
632
633     self._sort()
634
635     self._copy()
636
637     self._verify_integrity()
638
639     self._finalize()
640
641     self._sort()
642
643     self._copy()
644
645     self._verify_integrity()
646
647     self._finalize()
648
649     self._sort()
650
651     self._copy()
652
653     self._verify_integrity()
654
655     self._finalize()
656
657     self._sort()
658
659     self._copy()
660
661     self._verify_integrity()
662
663     self._finalize()
664
665     self._sort()
666
667     self._copy()
668
669     self._verify_integrity()
670
671     self._finalize()
672
673     self._sort()
674
675     self._copy()
676
677     self._verify_integrity()
678
679     self._finalize()
680
681     self._sort()
682
683     self._copy()
684
685     self._verify_integrity()
686
687     self._finalize()
688
689     self._sort()
690
691     self._copy()
692
693     self._verify_integrity()
694
695     self._finalize()
696
697     self._sort()
698
699     self._copy()
700
701     self._verify_integrity()
702
703     self._finalize()
704
705     self._sort()
706
707     self._copy()
708
709     self._verify_integrity()
710
711     self._finalize()
712
713     self._sort()
714
715     self._copy()
716
717     self._verify_integrity()
718
719     self._finalize()
720
721     self._sort()
722
723     self._copy()
724
725     self._verify_integrity()
726
727     self._finalize()
728
729     self._sort()
730
731     self._copy()
732
733     self._verify_integrity()
734
735     self._finalize()
736
737     self._sort()
738
739     self._copy()
740
741     self._verify_integrity()
742
743     self._finalize()
744
745     self._sort()
746
747     self._copy()
748
749     self._verify_integrity()
750
751     self._finalize()
752
753     self._sort()
754
755     self._copy()
756
757     self._verify_integrity()
758
759     self._finalize()
760
761     self._sort()
762
763     self._copy()
764
765     self._verify_integrity()
766
767     self._finalize()
768
769     self._sort()
770
771     self._copy()
772
773     self._verify_integrity()
774
775     self._finalize()
776
777     self._sort()
778
779     self._copy()
780
781     self._verify_integrity()
782
783     self._finalize()
784
785     self._sort()
786
787     self._copy()
788
789     self._verify_integrity()
790
791     self._finalize()
792
793     self._sort()
794
795     self._copy()
796
797     self._verify_integrity()
798
799     self._finalize()
800
801     self._sort()
802
803     self._copy()
804
805     self._verify_integrity()
806
807     self._finalize()
808
809     self._sort()
810
811     self._copy()
812
813     self._verify_integrity()
814
815     self._finalize()
816
817     self._sort()
818
819     self._copy()
820
821     self._verify_integrity()
822
823     self._finalize()
824
825     self._sort()
826
827     self._copy()
828
829     self._verify_integrity()
830
831     self._finalize()
832
833     self._sort()
834
835     self._copy()
836
837     self._verify_integrity()
838
839     self._finalize()
840
841     self._sort()
842
843     self._copy()
844
845     self._verify_integrity()
846
847     self._finalize()
848
849     self._sort()
850
851     self._copy()
852
853     self._verify_integrity()
854
855     self._finalize()
856
857     self._sort()
858
859     self._copy()
860
861     self._verify_integrity()
862
863     self._finalize()
864
865     self._sort()
866
867     self._copy()
868
869     self._verify_integrity()
870
871     self._finalize()
872
873     self._sort()
874
875     self._copy()
876
877     self._verify_integrity()
878
879     self._finalize()
880
881     self._sort()
882
883     self._copy()
884
885     self._verify_integrity()
886
887     self._finalize()
888
889     self._sort()
890
891     self._copy()
892
893     self._verify_integrity()
894
895     self._finalize()
896
897     self._sort()
898
899     self._copy()
900
901     self._verify_integrity()
902
903     self._finalize()
904
905     self._sort()
906
907     self._copy()
908
909     self._verify_integrity()
910
911     self._finalize()
912
913     self._sort()
914
915     self._copy()
916
917     self._verify_integrity()
918
919     self._finalize()
920
921     self._sort()
922
923     self._copy()
924
925     self._verify_integrity()
926
927     self._finalize()
928
929     self._sort()
930
931     self._copy()
932
933     self._verify_integrity()
934
935     self._finalize()
936
937     self._sort()
938
939     self._copy()
940
941     self._verify_integrity()
942
943     self._finalize()
944
945     self._sort()
946
947     self._copy()
948
949     self._verify_integrity()
950
951     self._finalize()
952
953     self._sort()
954
955     self._copy()
956
957     self._verify_integrity()
958
959     self._finalize()
960
961     self._sort()
962
963     self._copy()
964
965     self._verify_integrity()
966
967     self._finalize()
968
969     self._sort()
970
971     self._copy()
972
973     self._verify_integrity()
974
975     self._finalize()
976
977     self._sort()
978
979     self._copy()
980
981     self._verify_integrity()
982
983     self._finalize()
984
985     self._sort()
986
987     self._copy()
988
989     self._verify_integrity()
990
991     self._finalize()
992
993     self._sort()
994
995     self._copy()
996
997     self._verify_integrity()
998
999     self._finalize()
1000
1001     self._sort()
1002
1003     self._copy()
1004
1005     self._verify_integrity()
1006
1007     self._finalize()
1008
1009     self._sort()
1010
1011     self._copy()
1012
1013     self._verify_integrity()
1014
1015     self._finalize()
1016
1017     self._sort()
1018
1019     self._copy()
1020
1021     self._verify_integrity()
1022
1023     self._finalize()
1024
1025     self._sort()
1026
1027     self._copy()
1028
1029     self._verify_integrity()
1030
1031     self._finalize()
1032
1033     self._sort()
1034
1035     self._copy()
1036
1037     self._verify_integrity()
1038
1039     self._finalize()
1040
1041     self._sort()
1042
1043     self._copy()
1044
1045     self._verify_integrity()
1046
1047     self._finalize()
1048
1049     self._sort()
1050
1051     self._copy()
1052
1053     self._verify_integrity()
1054
1055     self._finalize()
1056
1057     self._sort()
1058
1059     self._copy()
1060
1061     self._verify_integrity()
1062
1063     self._finalize()
1064
1065     self._sort()
1066
1067     self._copy()
1068
1069     self._verify_integrity()
1070
1071     self._finalize()
1072
1073     self._sort()
1074
1075     self._copy()
1076
1077     self._verify_integrity()
1078
1079     self._finalize()
1080
1081     self._sort()
1082
1083     self._copy()
1084
1085     self._verify_integrity()
1086
1087     self._finalize()
1088
1089     self._sort()
1090
1091     self._copy()
1092
1093     self._verify_integrity()
1094
1095     self._finalize()
1096
1097     self._sort()
1098
1099     self._copy()
1100
1101     self._verify_integrity()
1102
1103     self._finalize()
1104
1105     self._sort()
1106
1107     self._copy()
1108
1109     self._verify_integrity()
1110
1111     self._finalize()
1112
1113     self._sort()
1114
1115     self._copy()
1116
1117     self._verify_integrity()
1118
1119     self._finalize()
1120
1121     self._sort()
1122
1123     self._copy()
1124
1125     self._verify_integrity()
1126
1127     self._finalize()
1128
1129     self._sort()
1130
1131     self._copy()
1132
1133     self._verify_integrity()
1134
1135     self._finalize()
1136
1137     self._sort()
1138
1139     self._copy()
1140
1141     self._verify_integrity()
1142
1143     self._finalize()
1144
1145     self._sort()
1146
1147     self._copy()
1148
1149     self._verify_integrity()
1150
1151     self._finalize()
1152
1153     self._sort()
1154
1155     self._copy()
1156
1157     self._verify_integrity()
1158
1159     self._finalize()
1160
1161     self._sort()
1162
1163     self._copy()
1164
1165     self._verify_integrity()
1166
1167     self._finalize()
1168
1169     self._sort()
1170
1171     self._copy()
1172
1173     self._verify_integrity()
1174
1175     self._finalize()
1176
1177     self._sort()
1178
1179     self._copy()
1180
1181     self._verify_integrity()
1182
1183     self._finalize()
1184
1185     self._sort()
1186
1187     self._copy()
1188
1189     self._verify_integrity()
1190
1191     self._finalize()
1192
1193     self._sort()
1194
1195     self._copy()
1196
1197     self._verify_integrity()
1198
1199     self._finalize()
1200
1201     self._sort()
1202
1203     self._copy()
1204
1205     self._verify_integrity()
1206
1207     self._finalize()
1208
1209     self._sort()
1210
1211     self._copy()
1212
1213     self._verify_integrity()
1214
1215     self._finalize()
1216
1217     self._sort()
1218
1219     self._copy()
1220
1221     self._verify_integrity()
1222
1223     self._finalize()
1224
1225     self._sort()
1226
1227     self._copy()
1228
1229     self._verify_integrity()
1230
1231     self._finalize()
1232
1233     self._sort()
1234
1235     self._copy()
1236
1237     self._verify_integrity()
1238
1239     self._finalize()
1240
1241     self._sort()
1242
1243     self._copy()
1244
1245     self._verify_integrity()
1246
1247     self._finalize()
1248
1249     self._sort()
1250
1251     self._copy()
1252
1253     self._verify_integrity()
1254
1255     self._finalize()
1256
1257     self._sort()
1258
1259     self._copy()
1260
1261     self._verify_integrity()
1262
1263     self._finalize()
1264
1265     self._sort()
1266
1267     self._copy()
1268
1269     self._verify_integrity()
1270
1271     self._finalize()
1272
1273     self._sort()
1274
1275     self._copy()
1276
1277     self._verify_integrity()
1278
1279     self._finalize()
1280
1281     self._sort()
1282
1283     self._copy()
1284
1285     self._verify_integrity()
1286
1287     self._finalize()
1288
1289     self._sort()
1290
1291     self._copy()
1292
1293     self._verify_integrity()
1294
1295     self._finalize()
1296
1297     self._sort()
1298
1299     self._copy()
1300
1301     self._verify_integrity()
1302
1303     self._finalize()
1304
1305     self._sort()
1306
1307     self._copy()
1308
1309     self._verify_integrity()
1310
1311     self._finalize()
1312
1313     self._sort()
1314
1315     self._copy()
1316
1317     self._verify_integrity()
1318
1319     self._finalize()
1320
1321     self._sort()
1322
1323     self._copy()
1324
1325     self._verify_integrity()
1326
1327     self._finalize()
1328
1329     self._sort()
1330
1331     self._copy()
1332
1333     self._verify_integrity()
1334
1335     self._finalize()
1336
1337     self._sort()
1338
1339     self._copy()
1340
1341     self._verify_integrity()
1342
1343     self._finalize()
1344
1345     self._sort()
1346
1347     self._copy()
1348
1349     self._verify_integrity()
1350
1351     self._finalize()
1352
1353     self._sort()
1354
1355     self._copy()
1356
1357     self._verify_integrity()
1358
1359     self._finalize()
1360
1361     self._sort()
1362
1363     self._copy()
1364
1365     self._verify_integrity()
1366
1367     self._finalize()
1368
1369     self._sort()
1370
1371     self._copy()
1372
1373     self._verify_integrity()
1374
1375     self._finalize()
1376
1377     self._sort()
1378
1379     self._copy()
1380
1381     self._verify_integrity()
1382
1383     self._finalize()
1384
1385     self._sort()
1386
1387     self._copy()
1388
1389     self._verify_integrity()
1390
1391     self._finalize()
1392
1393     self._sort()
1394
1395     self._copy()
1396
1397     self._verify_integrity()
1398
1399     self._finalize()
1400
1401     self._sort()
1402
1403     self._copy()
1404
1405     self._verify_integrity()
1406
1407     self._finalize()
1408
1409     self._sort()
1410
1411     self._copy()
1412
1413     self._verify_integrity()
1414
1415     self._finalize()
1416
1417     self._sort()
1418
1419     self._copy()
1420
1421     self._verify_integrity()
1422
1423     self._finalize()
1424
1425     self._sort()
1426
1427     self._copy()
1428
1429     self._verify_integrity()
1430
1431     self._finalize()
1432
1433     self._sort()
1434
1435     self._copy()
1436
1437     self._verify_integrity()
1438
1439     self._finalize()
1440
1441     self._sort()
1442
1443     self._copy()
1444
1445     self._verify_integrity()
1446
1447     self._finalize()
1448
1449     self._sort()
1450
1451     self._copy()
1452
1453     self._verify_integrity()
1454
1455     self._finalize()
1456
1457     self._sort()
1458
1459     self._copy()
1460
1461     self._verify_integrity()
1462
1463     self._finalize()
1464
1465     self._sort()
1466
1467     self._copy()
1468
1469     self._verify_integrity()
1470
1471     self._finalize()
1472
1473     self._sort()
1474
1475     self._copy()
1476
1477     self._verify_integrity()
1478
1479     self._finalize()
1480
1481     self._sort()
1482
1483     self._copy()
1484
1485     self._verify_integrity()
1486
1487     self._finalize()
1488
1489     self._sort()
1490
1491     self._copy()
1492
1493     self._verify_integrity()
1494
1495     self._finalize()
1496
1497     self._sort()
1498
1499     self._copy()
1500
1501     self._verify_integrity()
1502
1503     self._finalize()
1504
1505     self._sort()
1506
1507     self._copy()
1508
1509     self._verify_integrity()
1510
1511     self._finalize()
1512
1513     self._sort()
1514
1515     self._copy()
1516
1517     self._verify_integrity()
1518
1519     self._finalize()
1520
1521     self._sort()
1522
1523     self._copy()
1524
1525     self._verify_integrity()
1526
1527     self._finalize()
1528
1529     self._sort()
1530
1531     self._copy()
1532
1533     self._verify_integrity()
1534
1535     self._finalize()
1536
1537     self._sort()
1538
1539     self._copy()
1540
1541     self._verify_integrity()
1542
1543     self._finalize()
1544
1545     self._sort()
1546
1547     self._copy()
1548
1549     self._verify_integrity()
1550
1551     self._finalize()
1552
1553     self._sort()
1554
1555     self._copy()
1556
1557     self._verify_integrity()
1558
1559     self._finalize()
1560
1561     self._sort()
1562
1563     self._copy()
1564
1565     self._verify_integrity()
1566
1567     self._finalize()
1568
1569     self._sort()
1570
1571     self._copy()
1572
1573     self._verify_integrity()
1574
1575     self._finalize()
1576
1577     self._sort()
1578
1579     self._copy()
1580
1581     self._verify_integrity()
1582
1583     self._finalize()
1584
1585     self._sort()
1586
1587     self._copy()
1588
1589     self._verify_integrity()
1590
1591     self._finalize()
1592
1593     self._sort()
1594
1595     self._copy()
1596
1597     self._verify_integrity()
1598
1599     self._finalize()
1600
1601     self._sort()
1602
1603     self._copy()
1604
1605     self._verify_integrity()
1606
1607     self._finalize()
1608
1609     self._sort()
1610
1611     self._copy()
1612
1613     self._verify_integrity()
1614
1615     self._finalize()
1616
1617     self._sort()
1618
1619     self._copy()
1620
1621     self._verify_integrity()
1622
1623     self._finalize()
1624
1625     self._sort()
1626
1627     self._copy()
1628
1629     self._verify_integrity()
1630
1631     self._finalize()
1632
1633     self._sort()
1634
1635     self._copy()
1636
1637     self._verify_integrity()
1638
1639     self._finalize()
1640
1641     self._sort()
1642
1643     self._copy()
1644
1645     self._verify_integrity()
1646
1647     self._finalize()
1648
1649     self._sort()
1650
1651     self._copy()
1652
1653     self._verify_integrity()
1654
1655     self._finalize()
1656
1657     self._sort()
1658
1659     self._copy()
1660
1661     self._verify_integrity()
1662
1663     self._finalize()
1664
1665     self._sort()
1666
1667     self._copy()
1668
1669     self._verify_integrity()
1670
1671     self._finalize()
1672
1673     self._sort()
1674
1675     self._copy()
1676
1677     self._verify_integrity()
1678
1679     self._finalize()
1680
1681     self._sort()
1682
1683     self._copy()
1684
1685     self._verify_integrity()
1686
1687     self._finalize()
1688
1689     self._sort()
1690
1691     self._copy()
1692
1693     self._verify_integrity()
1694
1695     self._finalize()
1696
1697     self._sort()
1698
1699     self._copy()
1700
1701     self._verify_integrity()
1702
1703     self._finalize()
1704
1705     self._sort()
1706
1707     self._copy()
1708
1709     self._verify_integrity()
1710
1711     self._finalize()
1712
1713     self._sort()
1714
1715     self._copy()
1716
1717     self._verify_integrity()
1718
1719     self._finalize()
1720
1721     self._sort()
1722
1723     self._copy()
1724
1725     self._verify_integrity()
1726
1727     self._finalize()
1728
1729     self._sort()
1730
1731     self._copy()
1732
1733     self._verify_integrity()
1734
1735     self._finalize()
1736
1737     self._sort()
1738
1739     self._copy()
1740
1741     self._verify_integrity()
1742
1743     self._finalize()
1744
1745     self._sort()
1746
1747     self._copy()
1748
1749     self._verify_integrity()
1750
1751     self._finalize()
1752
1753     self._sort()
1754
1755     self._copy()
1756
1757     self._verify_integrity()
1758
1759     self._finalize()
1760
1761     self._sort()
1762
1763     self._copy()
1764
1765     self._verify_integrity()
1766
1767     self._finalize()
1768
1769     self._sort()
1770
1771     self._copy()
1772
1773     self._verify_integrity()
1774
1775     self._finalize()
1776
1777     self._sort()
1778
1779     self._copy()
1780
1781     self._verify_integrity()
1782
1783     self._finalize()
1784
1785     self._sort()
1786
1787     self._copy()
1788
1789     self._verify_integrity()
1790
1791     self._finalize()
1792
1793     self._sort()
1794
1795     self._copy()
1796
1797     self._verify_integrity()
1798
1799     self._finalize()
1800
1801     self._sort()
1802
1803     self._copy()
1804
1805     self._verify_integrity()
1806
1807     self._finalize()
1808
1809     self._sort()
1810
1811     self._copy()
1812
1813     self._verify_integrity()
1814
1815     self._finalize()
1816
1817     self._sort()
1818
1819     self._copy()
1820
1821     self._verify_integrity()
1822
1823     self._finalize()
1824
1825     self._sort()
1826
1827     self._copy()
1828
1829     self._verify_integrity()
1830
1831     self._finalize()
1832
1833     self._sort()
1834
1835     self._copy()
1836
1837     self._verify_integrity()
1838
1839     self._finalize()
1840
1841     self._sort()
1842
1843     self._copy()
1844
1845     self._verify_integrity()
1846
1847     self._finalize()
1848
1849     self._sort()
1850
1851     self._copy()
1852
1853     self._verify_integrity()
1854
1855     self._finalize()
1856
1857     self._sort()
1858
1859     self._copy()
1860
1861     self._verify_integrity()
1862
1863     self._finalize()
1864
1865     self._sort()
1866
1867     self._copy()
1868
1869     self._verify_integrity()
1870
1871     self._finalize()
1872
1873     self._sort()
1874
1875     self._copy()
1876
1877     self._verify_integrity()
1878
1879     self._finalize()
1880
1881     self._sort()
1882
1883     self._copy()
1884
1885     self._verify_integrity()
1886
1887     self._finalize()
1888
1889     self._sort()
1890
1891     self._copy()
1892
1893     self._verify_integrity()
1894
1895     self._finalize()
1896
1897     self._sort()
1898
1899     self._copy()
1900
1901     self._verify_integrity()
1902
1903     self._finalize()
1904
1905     self._sort()
1906
1907     self._copy()
1908
1909     self._verify_integrity()
1910
1911     self._finalize()
1912
1913     self._sort()
1914
1915     self._copy()
1916
1917     self._verify_integrity()
1918
1919     self._finalize()
1920
1921     self._sort()
1922
1923     self._copy()
1924
1925     self._verify_integrity()
1926
1927     self._finalize()
1928
1929     self._sort()
1930
1931     self._copy()
1932
1933     self._verify_integrity()
1934
1935     self._finalize()
1936
1937     self._sort()
1938
1939     self._copy()
1940
1941     self._verify_integrity()
1942
1943     self._finalize()
1944
1945     self._sort()
1946
1947     self._copy()
1948
1949     self._verify_integrity()
1950
1951     self._finalize()
1952
1953     self._sort()
1954
1955     self._copy()
1956
1957     self._verify_integrity()
1958
1959     self._finalize()
1960
1961     self._sort()
1962
1963     self._copy()
1964
1965     self._verify_integrity()
1966
1967     self._finalize()
1968
1969     self._sort()
1970
1971     self._copy()
1972
1973     self._verify_integrity()
1974
1975     self._finalize()
1976
1977     self._sort()
1978
1979     self._copy()
1980
1981     self._verify_integrity()
1982
1983     self._finalize()
1984
1985     self._sort()
1986
1987     self._copy()
1988
1989     self._verify_integrity()
1990
1991     self._finalize()
1992
1993     self._sort()
1994
1995     self._copy()
1996
1997     self._verify_integrity()
1998
1999     self._finalize()
2000
2001     self._sort()
2002
2003     self._copy()
2004

```

#	Column	Non-Null Count	Dtype
0	X	31343 non-null	float64
1	Y	31343 non-null	float64
2	BldgNo	31343 non-null	int64
3	UnitNo	31343 non-null	int64
4	PostalCode	31343 non-null	int64
5	Location	31343 non-null	int64
6	Ward	31343 non-null	int64
7	NAICSCode	31343 non-null	int64
8	NAICSCat	31343 non-null	int64
9	Phone	31343 non-null	int64
10	Fax	31343 non-null	int64
11	TollFree	31343 non-null	int64
12	EMail	31343 non-null	int64
13	WebAddress	31343 non-null	int64
14	EmplRange	31343 non-null	int64
15	CENT_X	31343 non-null	float64
16	CENT_Y	31343 non-null	float64
17	isnew	31343 non-null	int64
18	Closed	31343 non-null	int64

dtypes: float64(4), int64(15)

memory usage: 4.8 MB

```
[114]: data.head()
```

```
[114]:
```

	X	Y	BldgNo	UnitNo	PostalCode	Location	Ward	\
46689	-79.665386	43.684736	0	0	0	0	5	
46690	-79.642760	43.593515	1	0	1	1	4	
46691	-79.667311	43.682752	1	0	0	0	5	
46692	-79.629235	43.698932	1	0	1	2	4	
46693	-79.629235	43.698932	1	0	1	2	4	

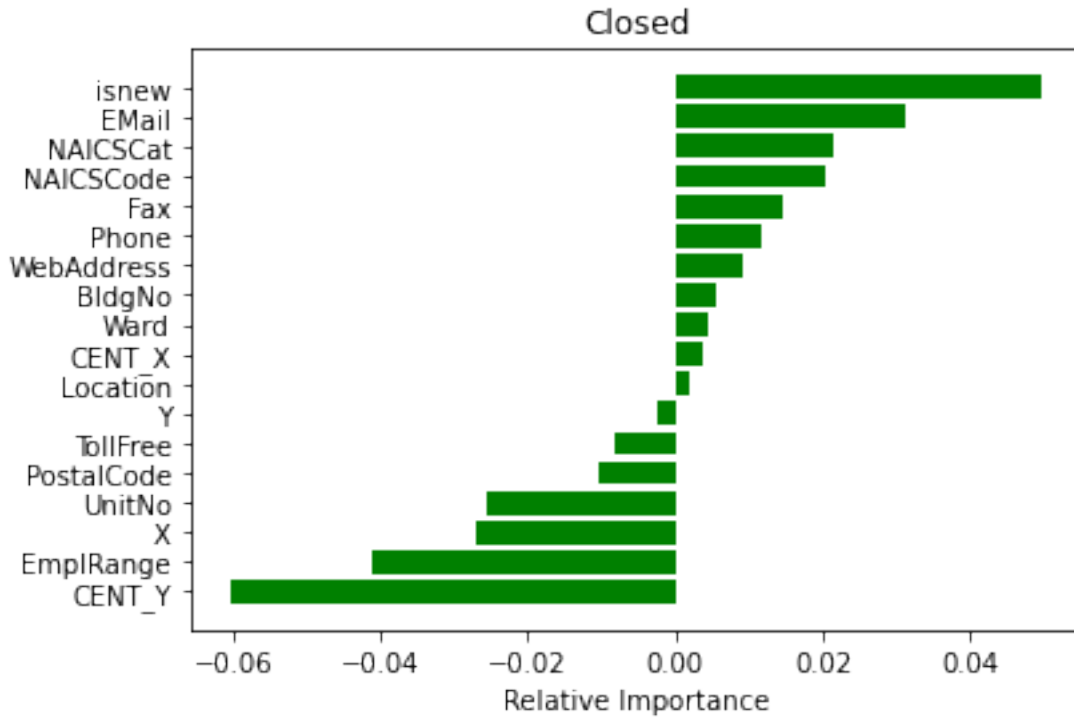
	NAICSCode	NAICSCat	Phone	Fax	TollFree	EMail	WebAddress	\
46689	0	0	0	0	0	0	0	
46690	1	1	1	1	1	1	0	
46691	2	2	0	0	0	0	0	
46692	0	0	0	1	1	1	0	
46693	0	0	0	1	1	1	1	

	EmplRange	CENT_X	CENT_Y	isnew	Closed
46689	1	607567.2334	4.837723e+06	0	0
46690	2	609556.5032	4.827621e+06	1	0
46691	3	607415.6044	4.837500e+06	0	0
46692	2	610454.8654	4.839347e+06	1	0
46693	4	610454.8654	4.839347e+06	1	0

```
[115]: importances = data.drop('Closed', axis=1).apply(lambda x: x.corr(data.Closed))
indices = np.argsort(importances)
print(importances[indices])
```

```
CENT_Y      -0.060260
EmplRange    -0.041116
X            -0.026877
UnitNo       -0.025721
PostalCode   -0.010325
TollFree     -0.008072
Y            -0.002403
Location      0.001719
CENT_X       0.003526
Ward         0.004591
BldgNo       0.005347
WebAddress   0.009212
Phone        0.011712
Fax          0.014476
NAICSCode    0.020264
NAICSCat     0.021286
EMail        0.031282
isnew        0.049747
dtype: float64
```

```
[116]: names=['X', 'Y', 'BldgNo', 'UnitNo', 'PostalCode', 'Location', 'Ward', 'NAICSCode', 'NAICSCat', 'Phone']
plt.title('Closed')
plt.barh(range(len(indices)), importances[indices], color='g', align='center')
plt.yticks(range(len(indices)), [names[i] for i in indices])
plt.xlabel('Relative Importance')
plt.show()
```



```
[136]: for i in range(0, len(indices)):
        if np.abs(importances[i])>0.04:
            print(names[i])
        #nothing is greater than .4 only 0.04! So nothing is correlated to Closed.
        ↪So all the code after this will yield nothing of significance.
```

```
EmplRange
CENT_Y
isnew
```

```
[118]: X= data[ ['EmplRange','CENT_Y','isnew']]
```

```
[119]: for i in range(0,len(X.columns)):
        for j in range(0,len(X.columns)):
            if i!=j:
                corr_1=np.abs(X[X.columns[i]].corr(X[X.columns[j]]))
                if corr_1 <0.3:
                    print( X.columns[i] , " is not correlated with ", X.columns[j])
                elif corr_1>0.75:
                    print( X.columns[i] , " is highly correlated with ", X.
                    ↪columns[j])
```

```
EmplRange is not correlated with CENT_Y
EmplRange is not correlated with isnew
```



```
CENT_Y is not correlated with EmplRange
CENT_Y is not correlated with isNew
isNew is not correlated with EmplRange
isNew is not correlated with CENT_Y
```

```
[132]: X = data[ ['EmplRange']]
```

```
[133]: y = data['Closed']
```

```
[134]: mi = mutual_info_regression(X, y)
```

```
[135]: mi = pd.Series(mi)
mi.index = X.columns
mi.sort_values(ascending=False)
mi.sort_values(ascending=False).plot.bar(figsize=(10, 4))
```

```
[135]: <Axes: >
```

