hw11

April 7, 2021

1 3220200915

2 Oakland Crime Statistics 2011 to 2016

2.0.1 Github https://github.com/lucien1998/DataMingLDY

[7]:
```
import seaborn as sns
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.ensemble import RandomForestRegressor
from fancyimpute import KNN
crime_data = pd.read_csv('records-for-2011.csv')
```

3 3.1

3.1 3.1.1

3.1.1 3.1.1 1

[8]:
```
# Agency(     )
print(crime_data['Agency'].value_counts())
```

```
OP    180015
Name: Agency, dtype: int64
```

[9]:
```
# Location(     )
print(crime_data['Location'].value_counts())
```

```
 INTERNATIONAL BLVD          3866
 MACARTHUR BLVD              3129
 AV&INTERNATIONAL BLVD       3067
 BROADWAY                    2132
 FOOTHILL BLVD               1791
                             …
```

```
57TH 19TH AV                   1
MACARTHUR COLLEGE AV           1
APGAR HARRISON ST              1
PIEDMONT 29TH AV               1
EDGEWATER WILSHIRE BLVD        1
Name: Location, Length: 32505, dtype: int64
```

[10]:

```
# Beat(     )
print(crime_data['Beat'].value_counts())
```

```
04X     7410
08X     6885
26Y     5478
30Y     5295
06X     5119
23X     5051
30X     4956
19X     4955
34X     4673
29X     4483
20X     4287
27Y     4159
07X     4134
31Y     4082
25X     4022
35X     3880
33X     3849
03X     3819
32X     3711
27X     3703
09X     3630
21Y     3435
32Y     3125
22X     3061
26X     2978
02Y     2970
10X     2967
14X     2733
03Y     2726
22Y     2664
12Y     2651
05X     2633
02X     2614
31X     2603
21X     2593
17Y     2582
24Y     2575
13Z     2546
```

```
15X     2509
24X     2459
12X     2422
10Y     2383
01X     2210
28X     2191
17X     2133
11X     2087
13Y     2017
35Y     1956
31Z     1870
18Y     1778
16Y     1561
14Y     1492
25Y     1482
13X     1122
18X     1063
16X      994
05Y      710
PDT2      20
Name: Beat, dtype: int64
```

[11]:

```
# Incident Type Id(     )
print(crime_data['Incident Type Id'].value_counts())
```

```
933R      17348
911H      12817
SECCK     11393
415       10752
10851      7180
          …
12020         1
666           1
591           1
YELALT        1
140           1
Name: Incident Type Id, Length: 263, dtype: int64
```

[12]:

```
# Incident Type Description(     )
print(crime_data['Incident Type Description'].value_counts())
```

```
ALARM-RINGER              17348
911 HANG-UP               12817
SECURITY CHECK            11393
STOLEN VEHICLE             7180
415 UNKNOWN                6624
                          …
TICKET SCALPING               1
```

```
PLAYING BALL IN STRE        1
OBSTRUCTING JUSTICE-        1
CONSPIRACY COURT ORD        1
FLOOD                       1
Name: Incident Type Description, Length: 265, dtype: int64
```

3.1.2 3.1.1 2 points price

```
[13]: #
      crime_data.describe()
```

```
[13]:             Area Id       Priority
      count  179112.000000  180015.000000
      mean        1.740648       1.796111
      std         0.746468       0.402916
      min         1.000000       0.000000
      25%         1.000000       2.000000
      50%         2.000000       2.000000
      75%         2.000000       2.000000
      max         3.000000       2.000000
```

```
[14]: #
      crime_data.info()
```

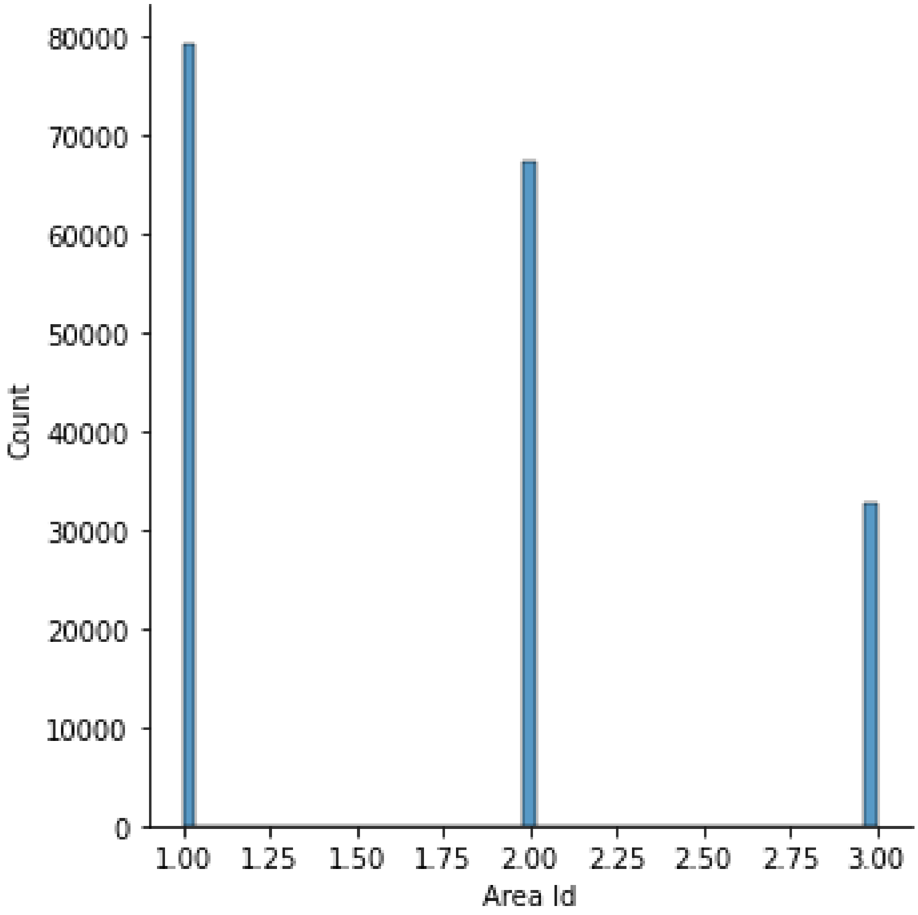
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180016 entries, 0 to 180015
Data columns (total 10 columns):
 #   Column                     Non-Null Count   Dtype
---  ------                     --------------   -----
 0   Agency                     180015 non-null  object
 1   Create Time                180015 non-null  object
 2   Location                   180016 non-null  object
 3   Area Id                    179112 non-null  float64
 4   Beat                       179496 non-null  object
 5   Priority                   180015 non-null  float64
 6   Incident Type Id           180015 non-null  object
 7   Incident Type Description  180015 non-null  object
 8   Event Number               180015 non-null  object
 9   Closed Time                180009 non-null  object
dtypes: float64(2), object(8)
memory usage: 13.7+ MB
```

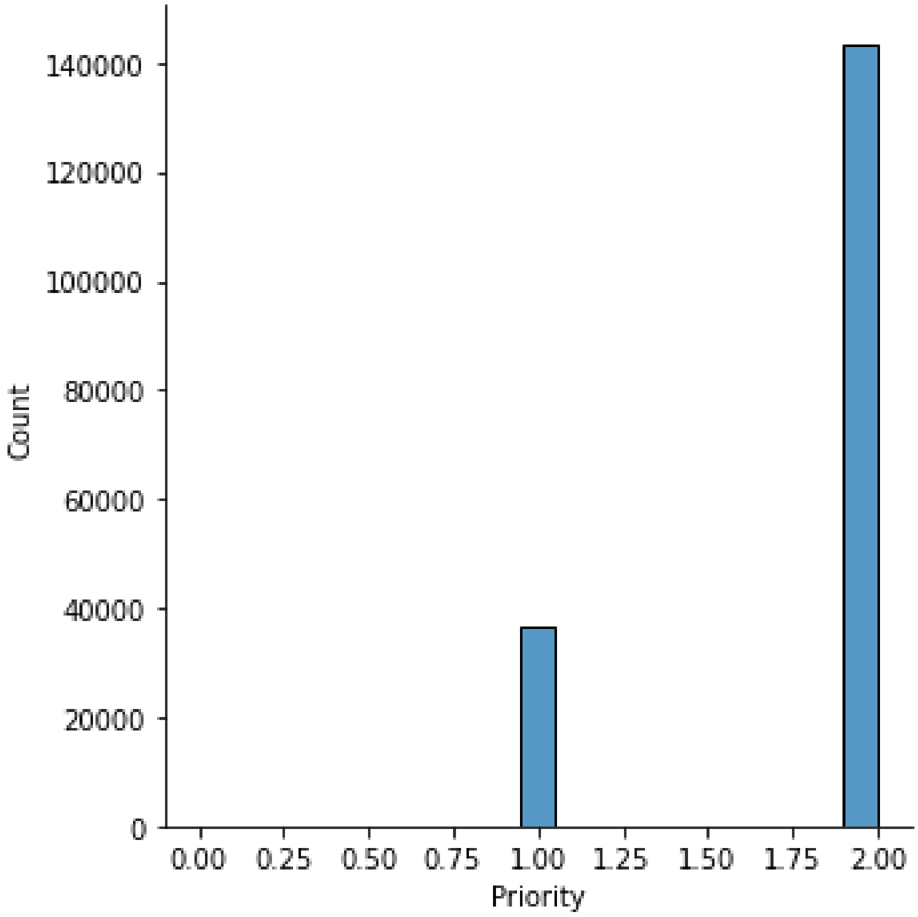
180015 Area ID 903 Beat 519 CloseTime 6 0

3.2 3.1.2

3.2.1 3.1.2 1

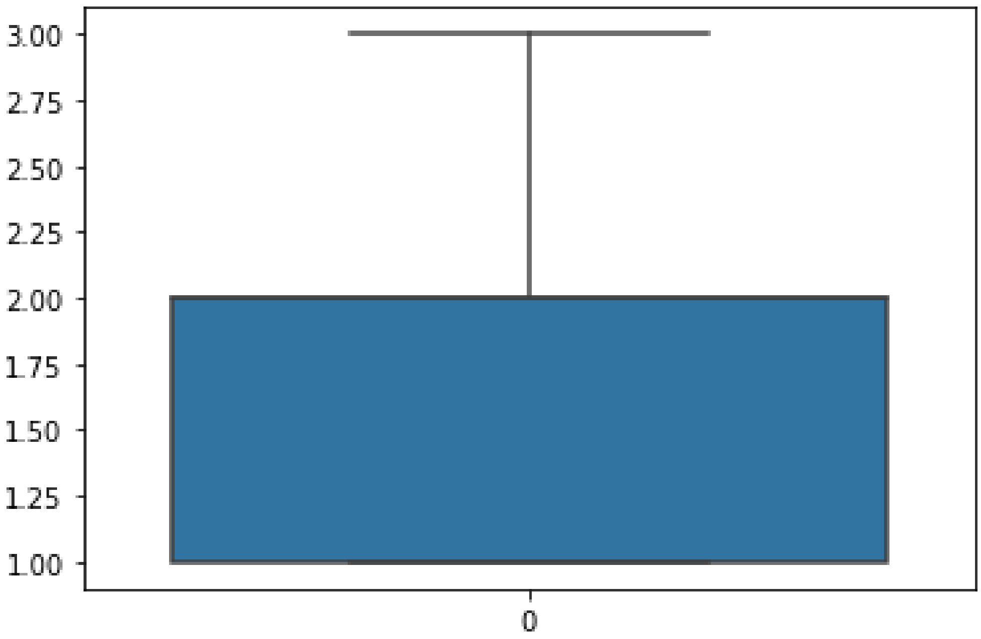
```
[15]: sns.displot(crime_data['Area Id'])
      plt.show()
      sns.displot(crime_data['Priority'])
      plt.show()
```

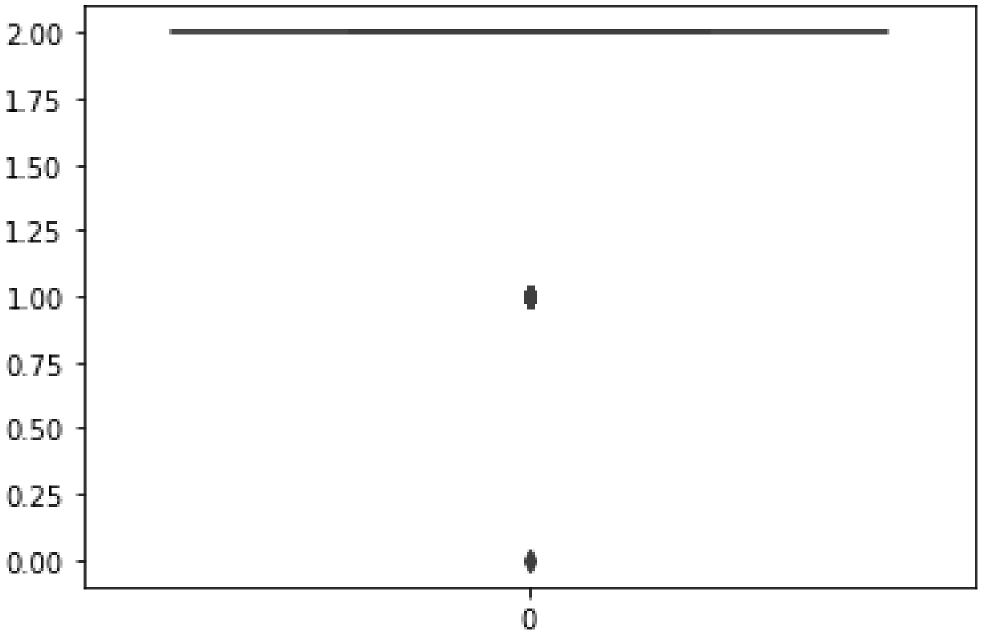




3.2.2 3.1.2 2

[16]:
```
sns.boxplot(data=crime_data['Area Id'])
plt.show()
sns.boxplot(data=crime_data['Priority'])
plt.show()
```





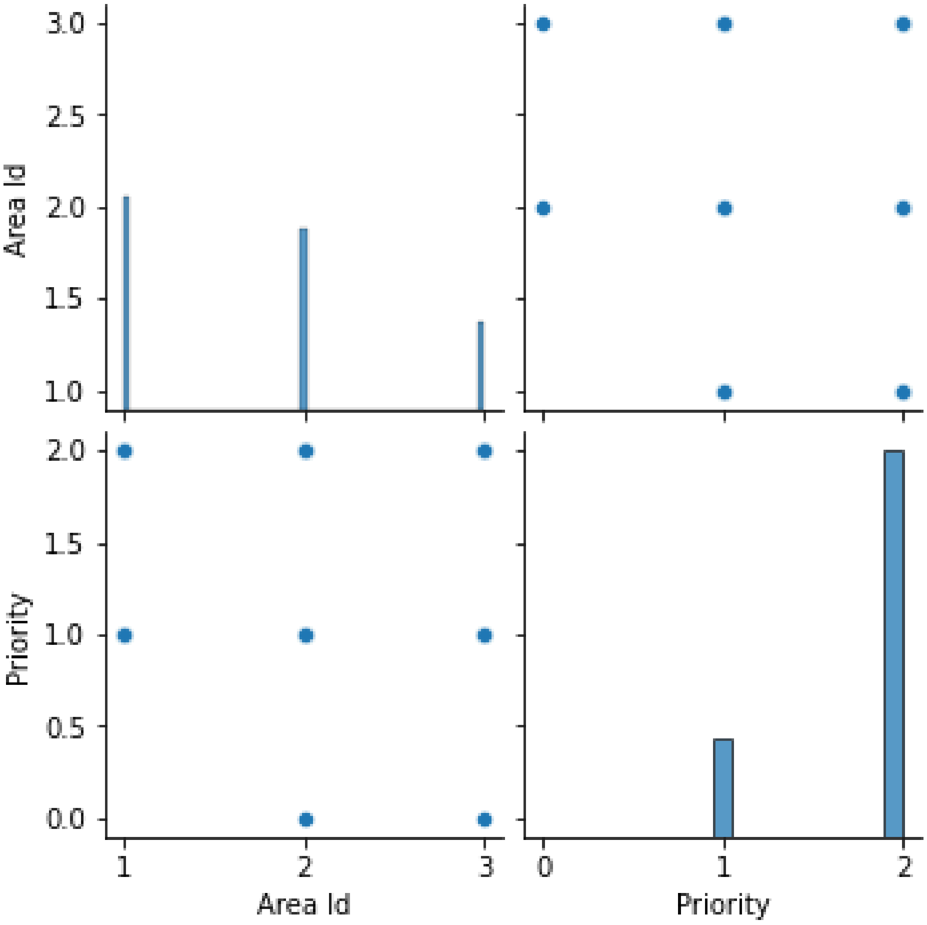
4 3.2

area_id area_id Beat

4.1 3.2.1

```
[17]: print(" 1  \n")
sns.pairplot(crime_data, vars=["Area Id", "Priority"])
plt.show()
print(crime_data['Area Id'])
print("-----------------------------------------------------------------------\n")
print(" 2  \n")
crime_data_after = crime_data.dropna()
sns.pairplot(crime_data_after, vars=["Area Id", "Priority"])
plt.show()
print(crime_data_after['Area Id'])
```

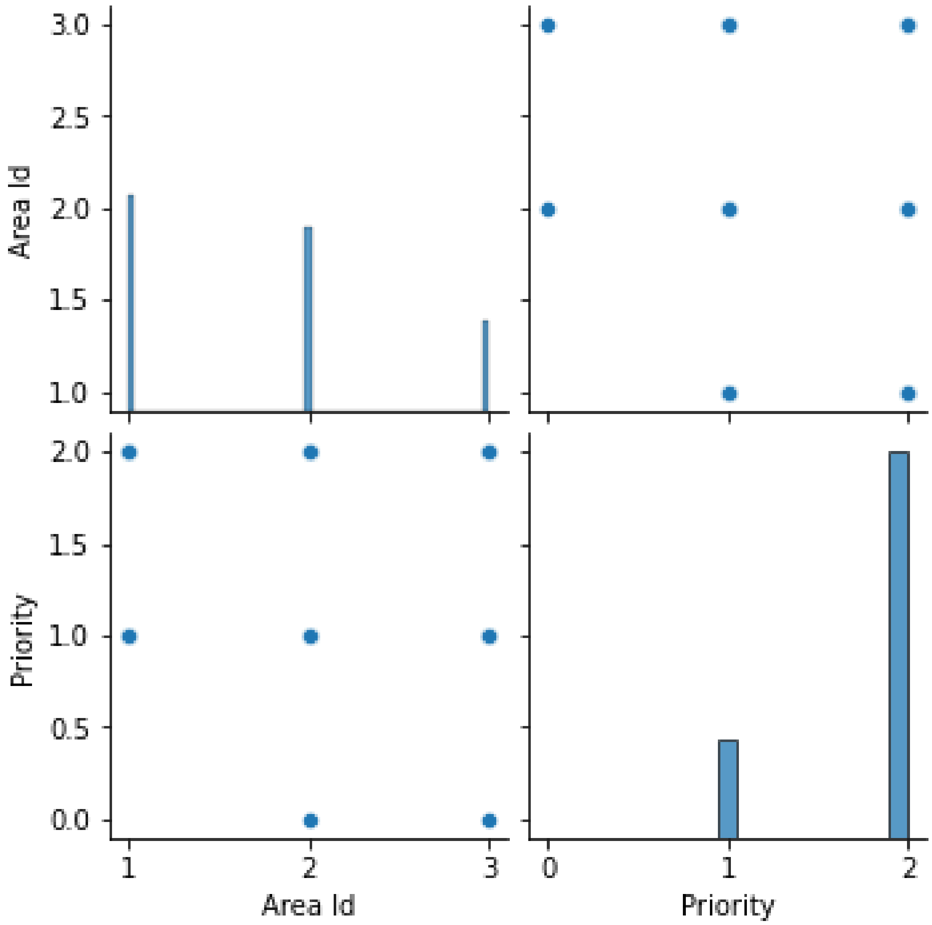
```
1
```



```
0          1.0
1          1.0
```

```
2          1.0
3          2.0
4          2.0
         …
180011    2.0
180012    1.0
180013    1.0
180014    2.0
180015    NaN
Name: Area Id, Length: 180016, dtype: float64
-----------------------------------------------------------------------
```

```
 2
```



```
0          1.0
1          1.0
2          1.0
```

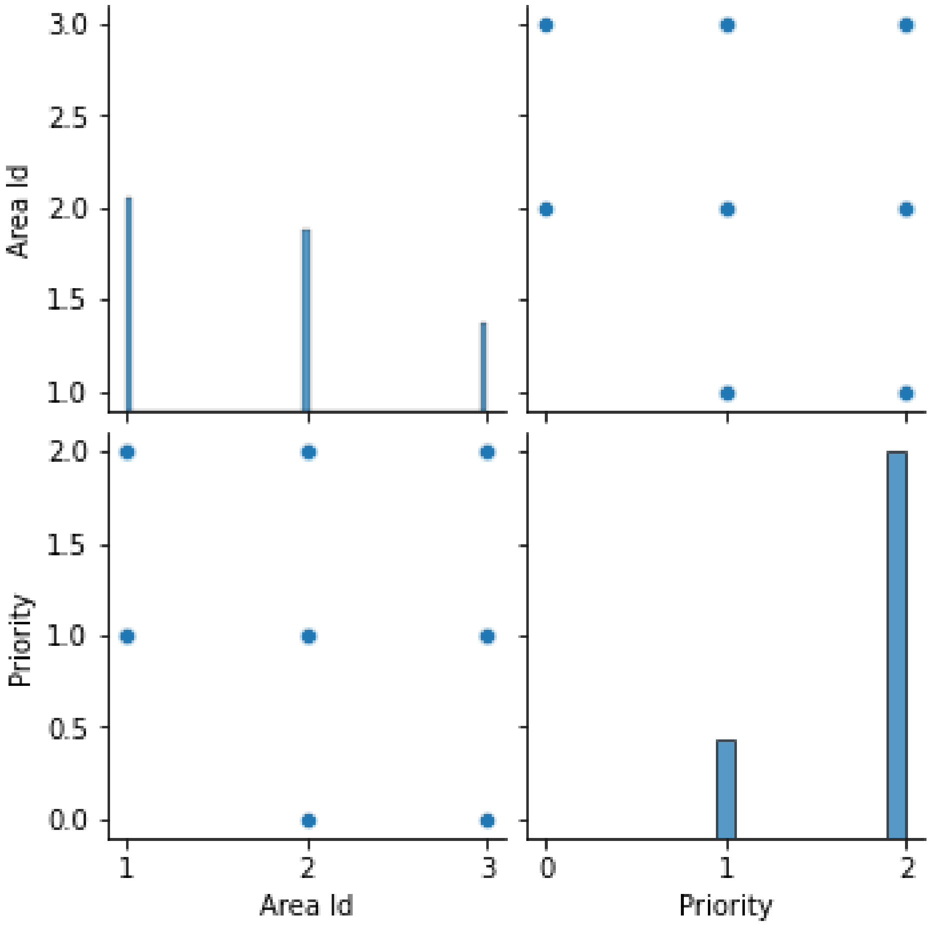
```
3         2.0
4         2.0
         …
180010    1.0
180011    2.0
180012    1.0
180013    1.0
180014    2.0
Name: Area Id, Length: 178771, dtype: float64
```

4.2 3.2.2

[27]:

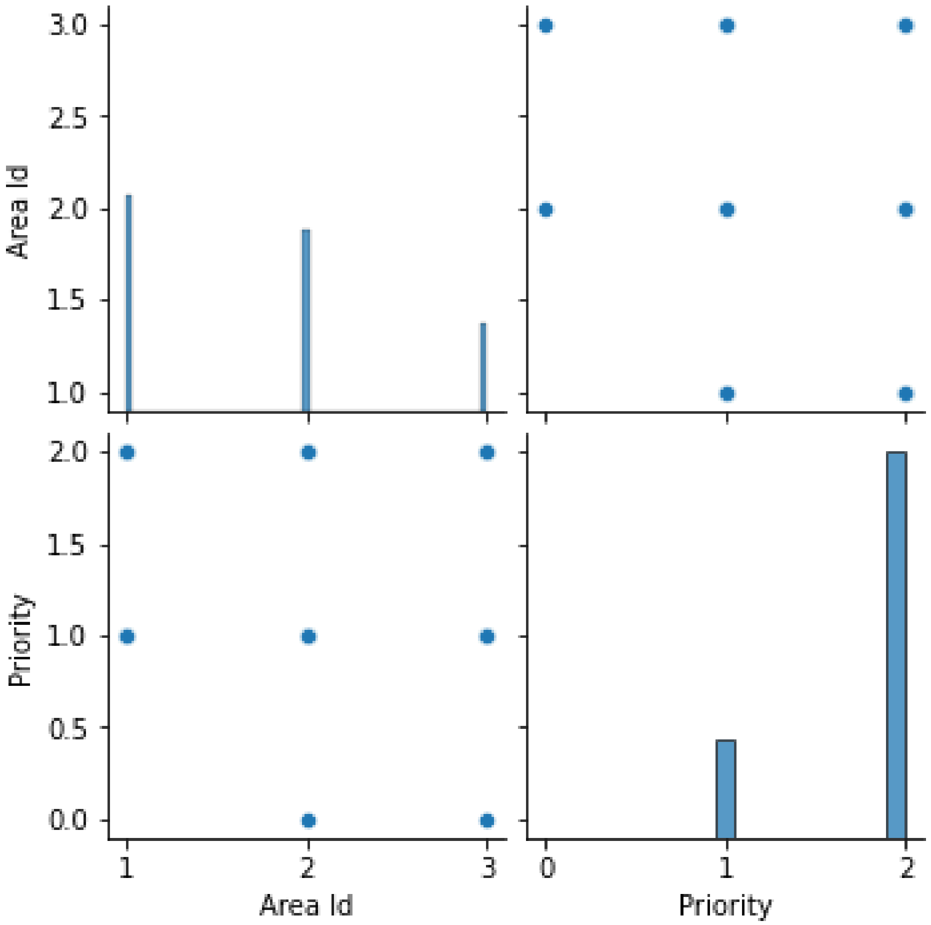
```
print(" 1  \n")
crime_data2 = crime_data.copy(deep=True)
sns.pairplot(crime_data2, vars=["Area Id","Priority"])
plt.show()
print(crime_data2['Area Id'])
print("\n===================\n")
crime_data2.info()
print("----------------------------------------------------------------------\n")
print(" 2  \n")
crime_data2['Area Id'].fillna(crime_data2['Area Id'].mode().
 ↪iloc[0],inplace=True)
crime_data2['Beat'].fillna(crime_data2['Beat'].mode().iloc[0],inplace=True)
sns.pairplot(crime_data2, vars=["Area Id","Priority"])
plt.show()
print(crime_data2['Area Id'])
print("\n===================\n")
crime_data2.info()
```

```
1
```



```
0         1.0
1         1.0
2         1.0
3         2.0
4         2.0
         …
180011    2.0
180012    1.0
180013    1.0
180014    2.0
180015    NaN
Name: Area Id, Length: 180016, dtype: float64

===================

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180016 entries, 0 to 180015
Data columns (total 10 columns):
 #   Column                     Non-Null Count   Dtype
---  ------                     --------------   -----
 0   Agency                     180015 non-null  object
 1   Create Time                180015 non-null  object
 2   Location                   180016 non-null  object
 3   Area Id                    179112 non-null  float64
 4   Beat                       179496 non-null  object
 5   Priority                   180015 non-null  float64
 6   Incident Type Id           180015 non-null  object
 7   Incident Type Description  180015 non-null  object
 8   Event Number               180015 non-null  object
 9   Closed Time                180009 non-null  object
dtypes: float64(2), object(8)
memory usage: 13.7+ MB
----------------------------------------------------------------------
```

```
 2
```



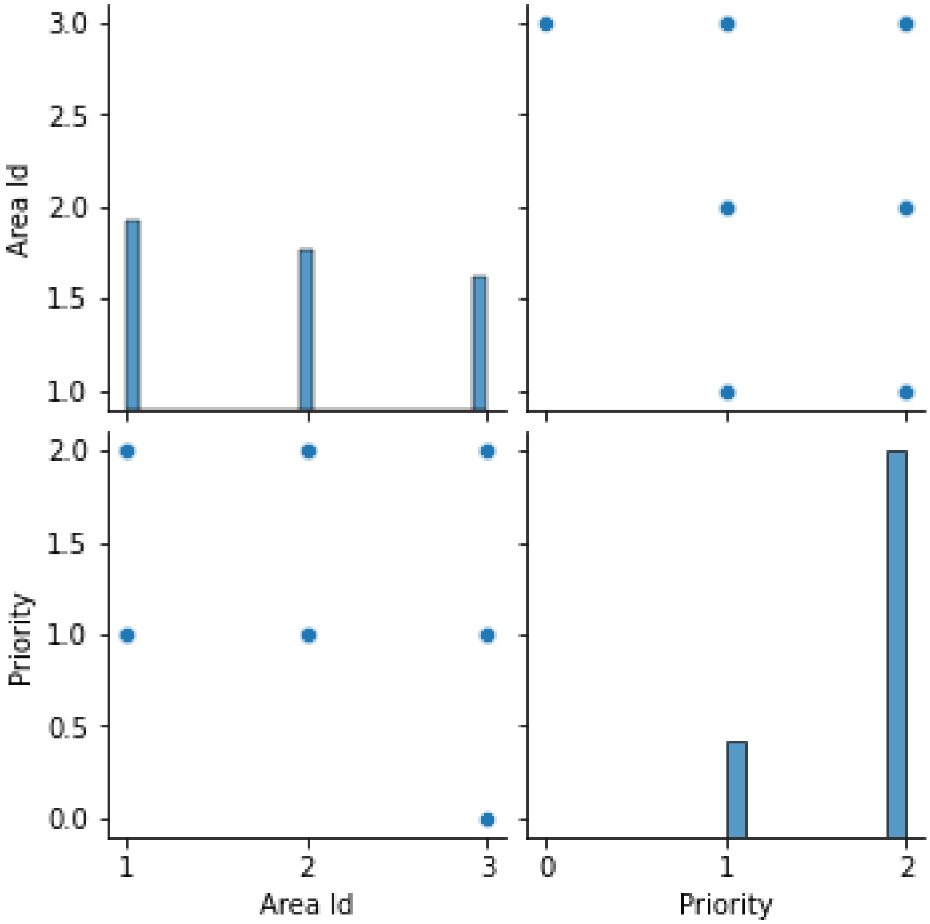
```
0         1.0
1         1.0
2         1.0
3         2.0
4         2.0
         …
180011    2.0
180012    1.0
180013    1.0
180014    2.0
180015    1.0
Name: Area Id, Length: 180016, dtype: float64

===================

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180016 entries, 0 to 180015
Data columns (total 10 columns):
 #   Column                     Non-Null Count   Dtype
---  ------                     --------------   -----
 0   Agency                     180015 non-null  object
 1   Create Time                180015 non-null  object
 2   Location                   180016 non-null  object
 3   Area Id                    180016 non-null  float64
 4   Beat                       180016 non-null  object
 5   Priority                   180015 non-null  float64
 6   Incident Type Id           180015 non-null  object
 7   Incident Type Description  180015 non-null  object
 8   Event Number               180015 non-null  object
 9   Closed Time                180009 non-null  object
dtypes: float64(2), object(8)
memory usage: 13.7+ MB
```

4.3 3.2.3

[36]:

```python
crime_dataY = pd.read_csv('records-for-2011-.csv')
crime_dataX = crime_dataY.copy(deep=True)
print(" 1  \n")
crime_data3 = crime_data.copy(deep=True)[:120000]
sns.pairplot(crime_data3, vars=["Area Id","Priority"])
plt.show()
print(crime_data3['Area Id'])
print("-----------------------------------------------------------------------\n")
print(" 2  \n")
def set_missing_AreaIds(df):
    #
    AreaId_df = df[['AreaId', 'Priority']]
    known_AreaId = AreaId_df[AreaId_df.AreaId.notnull()].iloc[:,:].values
    unknown_AreaId = AreaId_df[AreaId_df.AreaId.isnull()].iloc[:,:].values
    y = known_AreaId[:, 0]   # y AreaId
    x = known_AreaId[:, 1:]   # x
    rfr = RandomForestRegressor(random_state=0, n_estimators=2000, n_jobs=-1)
    #
    rfr.fit(x, y)
    #
    predictedAreaIds = rfr.predict(unknown_AreaId[:, 1:])
    #
    df.loc[(df.AreaId.isnull()), 'AreaId'] = predictedAreaIds
    return df
crime_dataX = set_missing_AreaIds(crime_dataX[:120000])
sns.pairplot(crime_dataX, vars=["AreaId","Priority"])
plt.show()
print(crime_dataX['AreaId'])
```

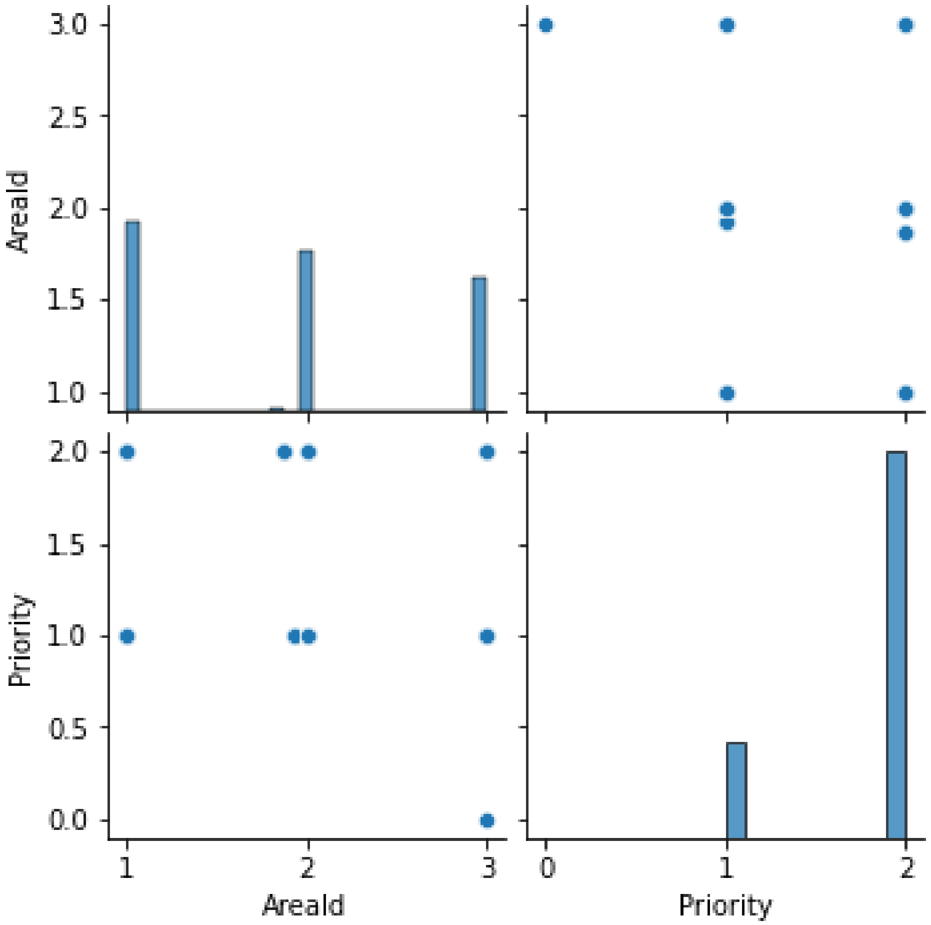
```
1
```



```
0         1.0
1         1.0
2         1.0
3         2.0
4         2.0
         …
119995    1.0
119996    1.0
119997    2.0
119998    1.0
119999    1.0
Name: Area Id, Length: 120000, dtype: float64
-----------------------------------------------------------------------
```

```
 2
```

```
d:\anaconda\envs\python373\lib\site-packages\pandas\core\indexing.py:1676:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  self._setitem_single_column(ilocs[0], value, pi)
```

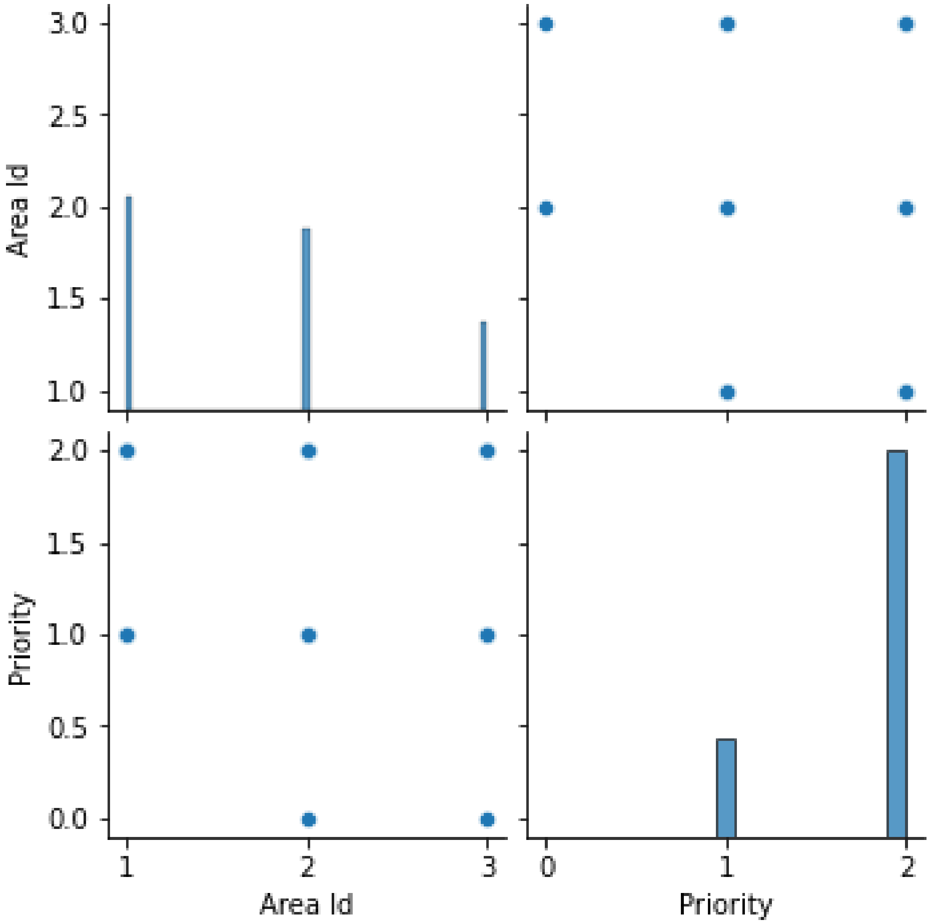


```
0          1.0
1          1.0
2          1.0
3          2.0
4          2.0
         …
119995    1.0
119996    1.0
119997    2.0
119998    1.0
119999    1.0
Name: AreaId, Length: 120000, dtype: float64
```

4.4 3.2.4

```
[21]: print(" 1  \n")
      crime_data4 = crime_data.copy(deep=True)
      sns.pairplot(crime_data4, vars=["Area Id","Priority"])
      plt.show()
      print(crime_data4['Area Id'])
      print("-----------------------------------------------------------------------\n")
      print(" 2  \n")
      new_data = crime_data4[['Area Id', 'Priority']][:10000]
      fill_knn = KNN(k=3).fit_transform(new_data)
      print(fill_knn)
```

```
1
```



```
0            1.0
1            1.0
```

```
2         1.0
3         2.0
4         2.0
         …
180011    2.0
180012    1.0
180013    1.0
180014    2.0
180015    NaN
Name: Area Id, Length: 180016, dtype: float64
-----------------------------------------------------------------------
```

```
 2
```

```
Imputing row 1/10000 with 0 missing, elapsed time: 10.648
Imputing row 101/10000 with 0 missing, elapsed time: 10.649
Imputing row 201/10000 with 0 missing, elapsed time: 10.650
Imputing row 301/10000 with 0 missing, elapsed time: 10.651
Imputing row 401/10000 with 0 missing, elapsed time: 10.652
Imputing row 501/10000 with 0 missing, elapsed time: 10.652
Imputing row 601/10000 with 0 missing, elapsed time: 10.653
Imputing row 701/10000 with 0 missing, elapsed time: 10.654
Imputing row 801/10000 with 0 missing, elapsed time: 10.654
Imputing row 901/10000 with 0 missing, elapsed time: 10.655
Imputing row 1001/10000 with 0 missing, elapsed time: 10.656
Imputing row 1101/10000 with 0 missing, elapsed time: 10.657
Imputing row 1201/10000 with 0 missing, elapsed time: 10.658
Imputing row 1301/10000 with 0 missing, elapsed time: 10.658
Imputing row 1401/10000 with 0 missing, elapsed time: 10.659
Imputing row 1501/10000 with 0 missing, elapsed time: 10.661
Imputing row 1601/10000 with 0 missing, elapsed time: 10.662
Imputing row 1701/10000 with 0 missing, elapsed time: 10.663
Imputing row 1801/10000 with 0 missing, elapsed time: 10.663
Imputing row 1901/10000 with 0 missing, elapsed time: 10.664
Imputing row 2001/10000 with 0 missing, elapsed time: 10.665
Imputing row 2101/10000 with 0 missing, elapsed time: 10.666
Imputing row 2201/10000 with 0 missing, elapsed time: 10.667
Imputing row 2301/10000 with 0 missing, elapsed time: 10.668
Imputing row 2401/10000 with 0 missing, elapsed time: 10.669
Imputing row 2501/10000 with 0 missing, elapsed time: 10.669
Imputing row 2601/10000 with 0 missing, elapsed time: 10.670
Imputing row 2701/10000 with 0 missing, elapsed time: 10.671
Imputing row 2801/10000 with 0 missing, elapsed time: 10.672
Imputing row 2901/10000 with 0 missing, elapsed time: 10.672
Imputing row 3001/10000 with 0 missing, elapsed time: 10.673
Imputing row 3101/10000 with 0 missing, elapsed time: 10.674
Imputing row 3201/10000 with 0 missing, elapsed time: 10.675
Imputing row 3301/10000 with 0 missing, elapsed time: 10.676
Imputing row 3401/10000 with 0 missing, elapsed time: 10.676
Imputing row 3501/10000 with 0 missing, elapsed time: 10.677
Imputing row 3601/10000 with 0 missing, elapsed time: 10.678
Imputing row 3701/10000 with 0 missing, elapsed time: 10.678
Imputing row 3801/10000 with 0 missing, elapsed time: 10.679
Imputing row 3901/10000 with 0 missing, elapsed time: 10.679
Imputing row 4001/10000 with 0 missing, elapsed time: 10.679
Imputing row 4101/10000 with 0 missing, elapsed time: 10.680
Imputing row 4201/10000 with 0 missing, elapsed time: 10.680
Imputing row 4301/10000 with 0 missing, elapsed time: 10.682
Imputing row 4401/10000 with 0 missing, elapsed time: 10.682
Imputing row 4501/10000 with 0 missing, elapsed time: 10.683
Imputing row 4601/10000 with 0 missing, elapsed time: 10.684
Imputing row 4701/10000 with 0 missing, elapsed time: 10.685
Imputing row 4801/10000 with 0 missing, elapsed time: 10.686
Imputing row 4901/10000 with 0 missing, elapsed time: 10.686
Imputing row 5001/10000 with 0 missing, elapsed time: 10.686
Imputing row 5101/10000 with 0 missing, elapsed time: 10.688
Imputing row 5201/10000 with 0 missing, elapsed time: 10.688
Imputing row 5301/10000 with 0 missing, elapsed time: 10.689
Imputing row 5401/10000 with 0 missing, elapsed time: 10.689
Imputing row 5501/10000 with 0 missing, elapsed time: 10.691
Imputing row 5601/10000 with 0 missing, elapsed time: 10.691
Imputing row 5701/10000 with 0 missing, elapsed time: 10.692
Imputing row 5801/10000 with 0 missing, elapsed time: 10.693
Imputing row 5901/10000 with 0 missing, elapsed time: 10.693
Imputing row 6001/10000 with 0 missing, elapsed time: 10.694
Imputing row 6101/10000 with 0 missing, elapsed time: 10.695
Imputing row 6201/10000 with 0 missing, elapsed time: 10.696
Imputing row 6301/10000 with 0 missing, elapsed time: 10.697
Imputing row 6401/10000 with 0 missing, elapsed time: 10.698
Imputing row 6501/10000 with 0 missing, elapsed time: 10.699
Imputing row 6601/10000 with 0 missing, elapsed time: 10.699
Imputing row 6701/10000 with 0 missing, elapsed time: 10.700
Imputing row 6801/10000 with 0 missing, elapsed time: 10.701
Imputing row 6901/10000 with 0 missing, elapsed time: 10.702
Imputing row 7001/10000 with 0 missing, elapsed time: 10.702
Imputing row 7101/10000 with 0 missing, elapsed time: 10.703
Imputing row 7201/10000 with 0 missing, elapsed time: 10.704
Imputing row 7301/10000 with 0 missing, elapsed time: 10.705
Imputing row 7401/10000 with 0 missing, elapsed time: 10.705
Imputing row 7501/10000 with 0 missing, elapsed time: 10.706
Imputing row 7601/10000 with 0 missing, elapsed time: 10.707
Imputing row 7701/10000 with 0 missing, elapsed time: 10.707
Imputing row 7801/10000 with 0 missing, elapsed time: 10.708
Imputing row 7901/10000 with 0 missing, elapsed time: 10.709
Imputing row 8001/10000 with 0 missing, elapsed time: 10.710
Imputing row 8101/10000 with 0 missing, elapsed time: 10.710
Imputing row 8201/10000 with 0 missing, elapsed time: 10.711
Imputing row 8301/10000 with 0 missing, elapsed time: 10.712
Imputing row 8401/10000 with 0 missing, elapsed time: 10.712
Imputing row 8501/10000 with 0 missing, elapsed time: 10.713
Imputing row 8601/10000 with 0 missing, elapsed time: 10.714
Imputing row 8701/10000 with 0 missing, elapsed time: 10.714
Imputing row 8801/10000 with 0 missing, elapsed time: 10.715
Imputing row 8901/10000 with 0 missing, elapsed time: 10.715
Imputing row 9001/10000 with 0 missing, elapsed time: 10.716
Imputing row 9101/10000 with 0 missing, elapsed time: 10.717
Imputing row 9201/10000 with 0 missing, elapsed time: 10.717
Imputing row 9301/10000 with 0 missing, elapsed time: 10.718
Imputing row 9401/10000 with 0 missing, elapsed time: 10.718
Imputing row 9501/10000 with 0 missing, elapsed time: 10.719
Imputing row 9601/10000 with 0 missing, elapsed time: 10.720
Imputing row 9701/10000 with 0 missing, elapsed time: 10.721
Imputing row 9801/10000 with 0 missing, elapsed time: 10.721
Imputing row 9901/10000 with 0 missing, elapsed time: 10.722
[[1. 1.]
 [1. 1.]
 [1. 2.]
 …
 [3. 2.]
 [1. 2.]
 [3. 1.]]
```