HW02 – Creating the Crime Dataset
Figures and Tables

	Subzone	Call_Number	Complaint	Date_Received	Day_Name	WEEK	MONTH	YEAR	YEAR_WEEK	SUB_YEAR_WEEK
1	ZONE1D	08C-000010	Burglary	2008-09-18 10:36:00	Thursday	38	9	2008	2008_38	2008_38_ZONE1D
2	ZONE2B	08C-000011	Burglary	2008-09-18 10:44:00	Thursday	38	9	2008	2008_38	2008_38_ZONE2B
3	ZONE1C	08C-000020	Assault	2008-09-18 11:52:00	Thursday	38	9	2008	2008_38	2008_38_ZONE1C
5	ZONE3D	08C-000029	Burglar alarm	2008-09-18 12:23:00	Thursday	38	9	2008	2008_38	2008_38_ZONE3D
6	ZONE4B	08C-000030	Welfare check	2008-09-18 12:26:00	Thursday	38	9	2008	2008_38	2008_38_ZONE4B

Figure 1: Screenshot of Calls with added columns including SUB_YEAR_WEEK

	SUB_YEAR_WEEK	call_Armed subject	call_Assault	call_Burglar alarm	call_Burglary	call_Disturbance	call_Domestic	call_FW FIREWORKS	call_Fight	call_Loitering	call_Message delivery
0	2008_38_ZONE1D	0	1	2	1	6	2	0	4	0	0
1	2008_39_ZONE1D	1	1	16	2	2	7	0	1	0	0
2	2008_40_ZONE1D	0	1	6	5	2	3	0	0	0	1
3	2008_41_ZONE1D	0	0	11	5	1	8	0	0	0	0
4	2008_42_ZONE1D	0	1	5	5	4	5	0	1	1	0

Figure 2: Screenshot of all calls aggregated by SUB_YEAR_WEEK

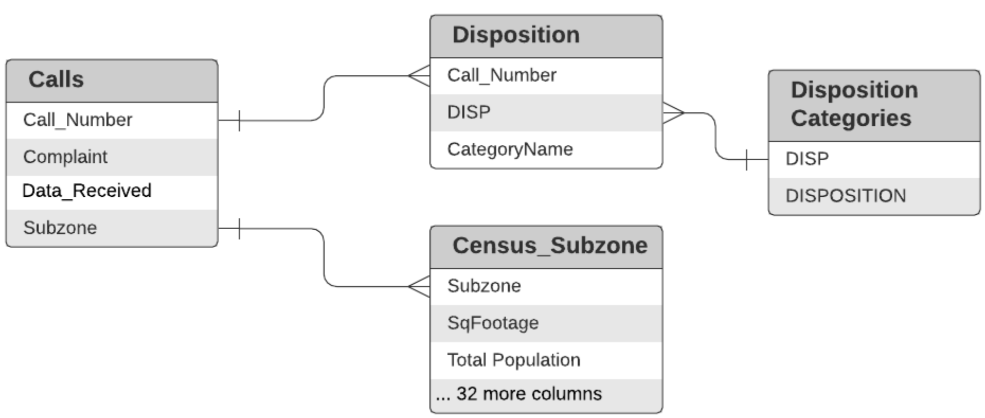

Figure 3: ERD for the data from the Canton Police Department

```
In [2]: from sqlalchemy import create_engine

In [3]: engine = create_engine('sqlite:///data/CantonPoliceDept.db')
        con = engine.connect()
        print('connection is ok')

        connection is ok

In [4]: #print(engine.table_names())

        from sqlalchemy import inspect

        insp = inspect(engine)
        print(insp.get_table_names())

        ['Calls', 'Disposition']

In [5]: rs = con.execute("SELECT * FROM Calls")
        df_calls = pd.DataFrame(rs.fetchall()) ##fetches all data from the Calls table
        df_calls.columns = rs.keys()
        display(df_calls.head(2))
        df_calls.info()
```

Figure 4: Screenshot of importing the Calls table from CantonPoliceDept.db

	Subzone	Call_Number	Complaint	Date_Received	Day_Name	WEEK	MONTH	YEAR	YEAR_WEEK	SUB_YEAR_WEEK
1	ZONE1D	08C-000010	Burglary	2008-09-18 10:36:00	Thursday	38	9	2008	2008_38	2008_38_ZONE1D
2	ZONE2B	08C-000011	Burglary	2008-09-18 10:44:00	Thursday	38	9	2008	2008_38	2008_38_ZONE2B
3	ZONE1C	08C-000020	Assault	2008-09-18 11:52:00	Thursday	38	9	2008	2008_38	2008_38_ZONE1C
5	ZONE3D	08C-000029	Burglar alarm	2008-09-18 12:23:00	Thursday	38	9	2008	2008_38	2008_38_ZONE3D
6	ZONE4B	08C-000030	Welfare check	2008-09-18 12:26:00	Thursday	38	9	2008	2008_38	2008_38_ZONE4B

Figure 5: Screenshot of df_calls

SUB_YEAR_WEEK	Friday	Monday	Saturday	Sunday	Thursday	Tuesday	Wednesday	month_1	month_2	month_3	month_4
2008_38_ZONE1D	0	0	0	0	1	0	0	0	0	0	0
2008_38_ZONE2B	0	0	0	0	1	0	0	0	0	0	0
2008_38_ZONE1C	0	0	0	0	1	0	0	0	0	0	0

Figure 6: Screenshot of df_calls with dummy variables for Day_Name and MONTH

```
df_calls_subweek_gr = df_calls.groupby('SUB_YEAR_WEEK')['WEEK'].count().reset_index()
df_calls_subweek_gr = df_calls_subweek_gr.rename(columns={'WEEK':'call_ALL'})
df_calls_subweek_gr
```

	SUB_YEAR_WEEK	call_ALL
0	2008_1_ZONE1A	12
1	2008_1_ZONE1B	9
2	2008_1_ZONE1C	16
3	2008_1_ZONE1D	17
4	2008_1_ZONE2A	8

Figure 7: Screenshot of df_calls_subweek_gr which create a sum of all calls per SUB_YEAR_WEEK

```
df_calls_subweek.head()
```

	SUB_YEAR_WEEK	Subzone	WEEK	MONTH	YEAR	YEAR_WEEK	call_ALL
0	2008_38_ZONE1D	ZONE1D	38	9	2008	2008_38	17
1	2008_38_ZONE2B	ZONE2B	38	9	2008	2008_38	1
2	2008_38_ZONE1C	ZONE1C	38	9	2008	2008_38	19
3	2008_38_ZONE3D	ZONE3D	38	9	2008	2008_38	11
4	2008_38_ZONE4B	ZONE4B	38	9	2008	2008_38	8

Figure 8: Screenshot of df_calls_subweek

	Call_Number	DISPOSITION	disp_15-ARREST	disp_66-CANCEL	disp_9-VERBAL	disp_ADV-ADVICE GIVEN	disp_AST-ASSIST	disp_CI-CITY CITATION	disp_CLR-CLEARED BY DISPATCH	disp_CM-CIVIL MATTER	disp
281571	16C-007256	RF-REPORT FILED	0	0	0	0	0	0	0	0	
858660	1115172	RES-RESOLVED	0	0	0	0	0	0	0	0	
14616	11C-047199	FA-FALSE ALARM	0	0	0	0	0	0	0	0	
595032	16C-099896	CLR-CLEARED BY DISPATCH	0	0	0	0	0	0	1	0	
531242	16C-102957	NR-NO REPORT	0	0	0	0	0	0	0	0	

Figure 9: Screenshot of df_disp

Table1: Columns and the Types of Conversion

Column	Conversion
Population_Male	Ratio based on 'Total Population'
Population_Female	Ratio based on 'Total Population'
Workers who travel to work	Ratio based on 'Worked'
Drove alone to Work	Ratio based on 'Worked'
Carpooled to Work	Ratio based on 'Worked'
Enrolled in school	Ratio based on 'Population_3andover'
Enrolled in nursery school, preschool	Ratio based on 'Population_3andover'
Enrolled in kindergarten	Ratio based on 'Population_3andover'
Enrolled in college, undergraduate years	Ratio based on 'Population_3andover'
Graduate or professional school	Ratio based on 'Population_3andover'
Not enrolled in school	Ratio based on 'Population_3andover'
Households_wageorsalaryincome	Ratio based on 'Households_earnings
Households_selfemploymentincome	Ratio based on 'Households_earnings
Households_interest_dividends	Ratio based on 'Households_earnings
Households_SSI	Ratio based on 'Households_earnings
Households_publicassistanceincome	Ratio based on 'Households_earnings

	SUB_YEAR_WEEK	Subzone	WEEK	MONTH	YEAR	YEAR_WEEK	call_ALL	Friday	Monday	Saturday	Sunday	Thursday
0	2008_38_ZONE1D	ZONE1D	38	9	2008	2008_38	17	4	0	8	2	3
1	2008_39_ZONE1D	ZONE1D	39	9	2008	2008_39	47	8	6	3	6	6
2	2008_40_ZONE1D	ZONE1D	40	9	2008	2008_40	35	8	5	1	6	5
3	2008_41_ZONE1D	ZONE1D	41	10	2008	2008_41	34	6	4	0	3	6
4	2008_42_ZONE1D	ZONE1D	42	10	2008	2008_42	36	4	4	9	5	4

Figure 10: Screenshot of df_calls_disp_week

	ZONE	ZONE5D	ZONE3X	ZONE1D	ZONE5B	ZONE6A	ZONE2C	ZONE3C	ZONE2B
0	ZONE5D	0.000000	61752.50885	27952.862110	19900.400120	53707.055550	35654.001300	40667.285820	46895.96176
1	ZONE3X	61752.508850	0.00000	38991.875170	41993.263080	13123.373080	45672.221490	21197.338720	51900.60212
2	ZONE1D	27952.862110	38991.87517	0.000000	11908.006760	28114.045530	13406.932120	19171.327440	25792.10749
3	ZONE5B	19900.400120	41993.26308	11908.006760	0.000000	33912.976880	24570.068460	20825.143480	37160.63945
4	ZONE6A	53707.055550	13123.37308	28114.045530	33912.976880	0.000000	32971.356490	14521.043490	38778.09899
5	ZONE2C	35654.001300	45672.22149	13406.932120	24570.068460	32971.356490	0.000000	29324.061470	12593.02489
6	ZONE3C	40667.285820	21197.33872	19171.327440	20825.143480	14521.043490	29324.061470	0.000000	39161.84481
7	ZONE2B	46895.961760	51900.60212	25792.107490	37160.639450	38778.098990	12593.024890	39161.844810	0.00000
8	ZONE7B	22525.868880	53691.31900	14787.102120	19157.881510	42309.235270	14362.005430	33867.268280	24467.10028
9	ZONE7D	7746.724534	57821.14229	21532.948730	16151.513120	48648.659030	28036.341130	36637.234190	39151.68497
10	ZONE4D	39803.548350	32676.64218	11871.281350	22763.474600	20025.292230	12997.801970	17965.751450	21227.73101
11	ZONE1A	32080.033990	32737.53815	6326.566288	13320.022370	22306.576810	17422.446330	12933.237220	28818.35915

Figure 11: Screenshot of df_spatial

	Subzone	Subzone_Comp	Subzone_Dist
0	ZONE5D	ZONE5D	0.00000
1	ZONE3X	ZONE5D	61752.50885
2	ZONE1D	ZONE5D	27952.86211
3	ZONE5B	ZONE5D	19900.40012
4	ZONE6A	ZONE5D	53707.05555

Figure 12: Screenshot of df_spatial_melt

	Subzone	Subzone_Comp	SUB_YEAR_WEEK	call_ALL	call_Burglary
0	ZONE7D	ZONE5D	2011_33_ZONE5D	1	0
1	ZONE7D	ZONE5D	2012_44_ZONE5D	1	0
2	ZONE7D	ZONE5D	2013_17_ZONE5D	2	0
3	ZONE7D	ZONE5D	2013_19_ZONE5D	1	0
4	ZONE7D	ZONE5D	2013_21_ZONE5D	1	0

Figure 13: Screenshot of df_spatial_burg

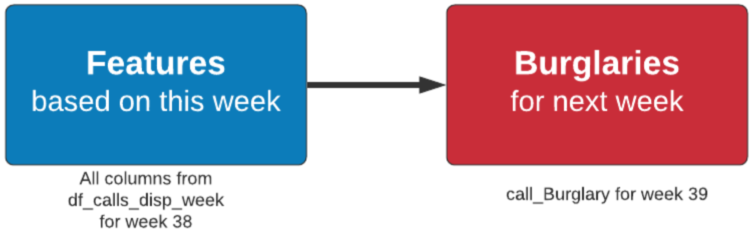

Figure 14: Example showing Time Lag between week 38 (current week) and week 39 (prediction week)

	SUB_YEAR_WEEK	Subzone	WEEK	MONTH	YEAR	YEAR_WEEK	call_ALL	call_Burglary	SUB_YEAR_WEEK_target	call_ALL_target	call_Burglary_target
0	2008_38_ZONE1D	ZONE1D	38	9	2008	2008_38	17	1	2008_39_ZONE1D	47	1
1	2008_38_ZONE2B	ZONE2B	38	9	2008	2008_38	1	1	2008_39_ZONE2B	2	0
2	2008_38_ZONE1C	ZONE1C	38	9	2008	2008_38	19	0	2008_39_ZONE1C	29	1
3	2008_38_ZONE3D	ZONE3D	38	9	2008	2008_38	11	0	2008_39_ZONE3D	20	2
4	2008_38_ZONE4B	ZONE4B	38	9	2008	2008_38	8	0	2008_39_ZONE4B	10	0

Figure 15: Screenshot of df_calls_subweek_target

	SUB_YEAR_WEEK	Subzone	WEEK	MONTH	YEAR	YEAR_WEEK	call_ALL	call_Burglary	SUB_YEAR_WEEK_target	call_ALL_target	call_Burglary_target
0	2008_38_ZONE1D	ZONE1D	38	9	2008	2008_38	17	1	2008_39_ZONE1D	47	1
22	2008_39_ZONE1D	ZONE1D	39	9	2008	2008_39	47	1	2008_40_ZONE1D	35	5
47	2008_40_ZONE1D	ZONE1D	40	9	2008	2008_40	35	5	2008_41_ZONE1D	34	3
64	2008_41_ZONE1D	ZONE1D	41	10	2008	2008_41	34	3	2008_42_ZONE1D	36	4
84	2008_42_ZONE1D	ZONE1D	42	10	2008	2008_42	36	4	2008_43_ZONE1D	38	2

Figure 16: Screenshot of df_calls_subweek_target filtered by Subzone = ZONE1D

```
df_pred_calls.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11868 entries, 0 to 11867
Columns: 116 entries, SUB_YEAR_WEEK to call_Burglary_comp
dtypes: UInt32(1), float64(20), int64(91), object(4)
memory usage: 10.6+ MB
```

```
df_pred_calls.head()
```

	SUB_YEAR_WEEK	SUB_YEAR_WEEK_target	call_ALL_target	call_Burglary_target	Subzone	WEEK	MONTH	YEAR	YEAR_WEEK	call_ALL	Friday	Monday
0	2008_38_ZONE1D	2008_39_ZONE1D	47	1	ZONE1D	38	9	2008	2008_38	17	4	0
1	2008_38_ZONE2B	2008_39_ZONE2B	2	0	ZONE2B	38	9	2008	2008_38	1	0	0
2	2008_38_ZONE1C	2008_39_ZONE1C	29	1	ZONE1C	38	9	2008	2008_38	19	4	0
3	2008_38_ZONE3D	2008_39_ZONE3D	20	2	ZONE3D	38	9	2008	2008_38	11	3	0
4	2008_38_ZONE4B	2008_39_ZONE4B	10	0	ZONE4B	38	9	2008	2008_38	8	4	0

```
df_pred_calls.columns
```

```
Index(['SUB_YEAR_WEEK', 'SUB_YEAR_WEEK_target', 'call_ALL_target',
       'call_Burglary_target', 'Subzone', 'WEEK', 'MONTH', 'YEAR', 'YEAR_WEEK',
       'call_ALL',
       ...
       'MedianAge_Total', 'MedianAge_Male', 'MedianAge_Female',
       'HouseholdIncome_Median', 'HouseholdIncome_Median_25to44',
       'HouseholdIncome_Median_65andover', 'HouseholdIncome_Median_45to64',
       'Income_PerCapita', 'call_ALL_comp', 'call_Burglary_comp'],
      dtype='object', length=116)
```

Figure 17: Screenshot of df_pred_calls

```
df_pred_calls[['SUB_YEAR_WEEK', 'SUB_YEAR_WEEK_target' ,'call_ALL_target',
               'call_Burglary_target', 'call_ALL_comp', 'call_Burglary_comp',
               'Burg_Status' ,'ALL_Status']].head()
```

	SUB_YEAR_WEEK	SUB_YEAR_WEEK_target	call_ALL_target	call_Burglary_target	call_ALL_comp	call_Burglary_comp	Burg_Status	ALL_Status
0	2008_38_ZONE1D	2008_39_ZONE1D	47	1	119.0	9.0	1	1
1	2008_38_ZONE2B	2008_39_ZONE2B	2	0	0.0	0.0	0	1
2	2008_38_ZONE1C	2008_39_ZONE1C	29	1	156.0	15.0	1	1
3	2008_38_ZONE3D	2008_39_ZONE3D	20	2	128.0	10.0	1	1
4	2008_38_ZONE4B	2008_39_ZONE4B	10	0	9.0	1.0	0	1

Figure 18: Screenshot of a select number of columns for the final df_pred_calls