

BIG DATA CRIME ANALYSIS FOR LOS ANGELES 2010-19

ΕΡΓΑΣΙΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΗΣ ΕΥΦΥΓΙΑΣ ΚΑΙ
ΑΝΑΛΥΣΗΣ ΜΕΓΑΛΩΝ ΔΕΔΟΜΕΝΩΝ
ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2020-21

Αθανασιάδης Αθανάσιος 8170002

Κυπριτίδης Στέφανος 8170050



Table of Contents

ΕΙΣΑΓΩΓΗ	4
DATASET	5
ETL	8
Data Extraction	8
Data Clearing	9
Data transforming into proper storage format.....	22
Removing and Renaming Columns.....	34
STAR SCHEMA.....	36
CUBE REPORTING.....	38
VISUALISATIONS.....	42
LA Crimes Overall	42
Crimes analysis based on sex of victim.....	43
Crimes analysis based on descent of victim	44
Crimes analysis based on age of the victim or suspect	44
Crime of rape	45
Battery – Simple Assault.....	46
Other	47
Crime visualization on map.....	48
Delay of report per crime type.....	49
Crimes percentages when suspect uses force.....	52
Crimes percentages when suspect is a stranger	54
DATA MINING	55
Clustering	55
Geographic coordinate clustering.....	55
Clustering on categorical variables.....	60
Clustering & examining different districts criminality.....	65
Prediction.....	72
Decision Trees	78
AdaBoost classifier.....	80
Bagging classifier	82
Random Forest.....	84

Extremely Randomized Trees	85
Συμπεράσματα.....	86
Πιθανότητες.....	88
ΕΠΙΛΟΓΟΣ.....	92
Βιβλιογραφία.....	93

ΕΙΣΑΓΩΓΗ

Ποτέ στο παρελθόν δεν είχαμε στην διάθεση μας τόσο πολλά δεδομένα. Δεδομένα συλλέγονται καθημερινά και από παντού. Συγκεκριμένα, για τα data sciences αυτό αποτελεί πολύ θετική εξέλιξη αφού δίνει την δυνατότητα για εξειδικευμένες αναλύσεις και εξόρυξη πολύτιμης γνώσης.

Η παρούσα ανάλυση αφορά την εγκληματικότητα και εστιάζει στην εύρεση ενός μεγάλου dataset, στην οπτικοποίηση του και στην εξόρυξη γνώσης χρησιμοποιώντας μοντέλα data mining. Αναλυτικά, η εργασία περιέχει τον καθαρισμό του dataset, την εισαγωγή του σε μια αποθήκη δεδομένων και την δημιουργία ενός κύβου δεδομένων και επιπλέον μετρικής. Επιπρόσθετα, παρουσιάζονται πολλές περιπτώσεις οπτικοποίησης δεδομένων όπου γίνονται πιο κατανοητά τα διάφορα δεδομένα. Άκομα, χρησιμοποιούνται οι λειτουργίες της συσταδοποίησης και της πρόβλεψης για την εξόρυξη σημαντικής γνώσης. Τέλος, όλα τα παραπάνω συνοδεύονται από πολλές εξηγήσεις και εικόνες για την πλήρη κατανόηση όλης της διαδικασίας της ανάλυσης.

DATASET

To dataset που χρησιμοποιήθηκε περιγράφει εγκλήματα που συνέβησαν στην περιοχή του Los Angeles την δεκαετία 2010-19. Συγκεκριμένα, αποτελεί κομμάτι ενός ευρύτερου dataset στο [Kaggle](#) που αφορά την αστυνομική βία και τις φυλετικές διαφορές αυτής. Το ευρύτερο dataset περιέχει διάφορες πληροφορίες από θανάτους πολιτών, αστυνομικών, δημογραφικά και στατιστικά εγκλημάτων έως και δεδομένα διαμαρτυριών (protests) και βία σε αυτές.

Επιλέξαμε να χρησιμοποιήσουμε το dataset με τα Los Angeles crimes για αρκετούς λόγους. Καταρχάς, στο συγκεκριμένο dataset, όπως θα δούμε μετά, υπάρχουν πολλές διαφορετικές ενδιαφέρουσες πληροφορίες για κάθε περιστατικό εγκλήματος. Αυτό είναι πολύ χρήσιμο και μας έδωσε την δυνατότητα να βγάλουμε συναρπαστικά συμπεράσματα και μοντέλα που θα αναλυθούν στην εργασία. Επιπρόσθετα, το γεγονός ότι αφορά πραγματικά περιστατικά και μάλιστα εγκλήματα αποτέλεσε κρίσιμο παράγοντα στην επιλογή του. Συγκεκριμένα, το dataset θα μπορούσε να αποτελεί real world problem για την μείωση και την αντιμετώπιση της εγκληματικότητας. Μάλιστα, το LAPD (Los Angeles Police Department) χρησιμοποιεί machine learning algorithms προκειμένου να προβλέπει εγκλήματα σε συγκεκριμένα τετράγωνα σε συγκεκριμένες χρονικές στιγμές. Τέλος, το Los Angeles συνιστά ελκυστική και πολυαγαπημένη πόλη, με αποτέλεσμα να μας τραβήξει το ενδιαφέρον.

Όσον αφορά το μέγεθος του αρχικού dataset, έχει περισσότερα από 2,1 εκατομμύρια εγγραφές (περιστατικά εγκλημάτων), 28 στήλες και συνολικό μέγεθος αρχείου περίπου 522 MB. Επιπρόσθετα, κάθε γραμμή του dataset αφορά ένα ξεχωριστό περιστατικό εγκλήματος που συνέβη στη περιοχή του Los Angeles από το 2010 έως και το 2019. Τώρα θα αναλυθούν οι διαφορετικές στήλες για κάθε καταγεγραμμένο περιστατικό εγκλήματος :

- DR_NO: επίσημος μοναδικός αριθμός εγγράφου που αποτελείται από ένα διψήφιο έτος, ένα αναγνωριστικό περιοχής και 5 ψηφία
- Date Rptd: σε μορφή MM/DD/YYYY, συνιστά τη χρονική στιγμή που έγινε η αναφορά του εγκλήματος
- DATE OCC: σε μορφή MM/DD/YYYY, συνιστά την ημερομηνία που πραγματοποιήθηκε το έγκλημα
- TIME OCC: σε μορφή HHMM, συνιστά την χρονική στιγμή (ώρα και λεπτά) που πραγματοποιήθηκε το έγκλημα.
- AREA: Το LAPD διαθέτει 21 κοινοτικούς Αστυνομικούς Σταθμούς που αναφέρονται ως Γεωγραφικές Περιοχές. Αυτές οι γεωγραφικές περιοχές αριθμούνται διαδοχικά από 1-21.
- AREA NAME: ονομασίες 21 γεωγραφικών περιοχών που αφορούν ένα ορόσημο ή τη γύρω κοινότητα της περιοχής.

- Rpt Dist No: τετραψήφιος κωδικός που αντιπροσωπεύει μια υποπεριοχή εντός μιας γεωγραφικής περιοχής.
- Part 1-2: κατηγορία εγκλήματος. Στην Αμερική οι παραβάσεις κατηγοριοποιούνται σε Part 1 offenses (e.g. murder, manslaughter, sex offenses) και Part 2 offenses (e.g., Weapon Violations, Prostitution)
- Crm Cd: υποδεικνύει το έγκλημα που διαπράχθηκε
- Crm Cd Desc: περιγράφει το Crm Cd ή αλλιώς η ονομασία του εγκλήματος που διαπράχθηκε
- Mocodes: δραστηριότητες που σχετίζονται με τον ύποπτο για τη διάπραξη του εγκλήματος
- Vict Age: η ηλικία του θύματος
- Vict Sex: το φύλο του θύματος:
 - F – Female
 - M – Male
 - X – Unknown
- Vict Descent: καταγωγή του θύματος όπου:
 - A - Other Asian
 - B - Black
 - C - Chinese
 - D - Cambodian
 - F - Filipino
 - G - Guamanian
 - H - Hispanic/Latin/Mexican
 - I - American Indian/Alaskan Native
 - J - Japanese
 - K - Korean
 - L - Laotian
 - O - Other
 - P - Pacific Islander
 - S - Samoan
 - U - Hawaiian
 - V - Vietnamese
 - W - White
 - X - Unknown
 - Z - Asian Indian
- Premis Cd: τύπος της δομής, του οχήματος ή της τοποθεσίας όπου συνέβη το έγκλημα
- Premis Desc: περιγραφή / ονομασία του Premis Cd
- Weapon Used Cd: ο τύπος όπλου που χρησιμοποιήθηκε στο περιστατικό
- Weapon Desc: περιγραφή/ ονομασία του Weapon Used Cd
- Status: κατάσταση του περιστατικού π.χ. έχει γίνει κάποια σύλληψη ή συνεχίζονται οι έρευνες.

- Status Desc: περιγραφή / ονομασία του Status
- Crm Cd 1: υποδεικνύει το έγκλημα που διαπράχθηκε. Crm Cd 1 είναι το πρωταρχικό και πιο σοβαρό.
- Crm Cd 2: μπορεί να περιέχει έναν κωδικό για ένα επιπλέον έγκλημα, λιγότερο σοβαρό από το Crm Cd 1
- Crm Cd 3: μπορεί να περιέχει έναν κωδικό για ένα επιπλέον έγκλημα, λιγότερο σοβαρό από το Crm Cd 2
- Crm Cd 4: μπορεί να περιέχει έναν κωδικό για ένα επιπλέον έγκλημα, λιγότερο σοβαρό από το Crm Cd 3
- LOCATION: οδός διεύθυνσης του περιστατικού εγκλήματος που έχει στρογγυλοποιηθεί στο πλησιέστερο εκατοστό τετράγωνο για να διατηρηθεί η ανωνυμία.
- Cross Street: διασταύρωση στρογγυλοποιημένης οδού
- LAT: γεωγραφικό πλάτος
- LON: γεωγραφικό μήκος

Σημαντική λεπτομέρεια καθιστά ότι αυτά τα δεδομένα μεταγράφονται από πρωτότυπες αναφορές εγκλημάτων που έχουν πληκτρολογηθεί σε χαρτί και, ως εκ τούτου, ενδέχεται να υπάρχουν ορισμένες ανακρίβειες στα δεδομένα. Ορισμένες εγγραφές έχουν missing data όσον αφορά την τοποθεσία και σημειώνονται ως (0° , 0°). Επίσης, τα πεδία διευθύνσεων παρέχονται μόνο στα πλησιέστερα εκατό μπλοκ για τη διατήρηση του απορρήτου. Τέλος, τα συγκεκριμένα δεδομένα είναι τόσο ακριβή όσο αυτά στη βάση δεδομένων του LAPD.

ETL

Data Extraction

Όσον αφορά την διαδικασία extract, κατεβάσαμε το συγκεκριμένο dataset από την ιστοσελίδα του [Kaggle](#) στους υπολογιστές μας.

Police Violence & Racial Equity - Part 2 of 3
Demographics, crime stats, and other data

JohnM • updated 3 days ago (Version 10)

Download (773 MB) New Notebook

Usability 10.0 License Other (specified in description) Tags social science, social issues and advocacy, demographics, racial equity

Description

This dataset is Part 2 of a three-part series that pulls together data from several different sources related to police violence and racial equity in the United States. The datasets currently include these types of data:

Part 1: Citizen deaths, police deaths, and other outcomes

- Police shootings
- Citizen fatalities involving police
- Police officer deaths suffered in the line of duty

Part 2: Demographics, crime stats, and other data

View Active Events

Data Explorer

3.36 GB

crime_data

- Chicago Crimes_20...
- Dallas Police Arrests.c...
- LA Crime_Data_from_...
- NYPD_Arrests_Data_...

LA Crime_Data_from_2010_to_2019.csv (510.09 MB)

About this file

Source: <https://data.lacity.org/A-Safe-City/Crime-Data-from-2020-to-Present/2nrs-mtv8>

Data Clearing

Στη συνέχεια, ακολούθησε η διαδικασία transform δηλαδή ο καθαρισμός δεδομένων και μετατροπή τους σε κατάλληλη μορφή αποθήκευσης για τους σκοπούς της έρευνας και της ανάλυσης. Για τη συγκεκριμένη διαδικασία χρησιμοποιήθηκαν τα εργαλεία pandas, numpy και jupyter notebook. Συγκεκριμένα, ο καθαρισμός πραγματοποιήθηκε μέσα στο περιβάλλον του jupyter notebook, χρησιμοποιώντας τις βιβλιοθήκες της Python ονόματι pandas και numpy. Οι συγκεκριμένες βιβλιοθήκες συνιστούν από τις πιο ευρέως χρησιμοποιούμενες βιβλιοθήκες της python όσον αφορά τα data sciences.

Τώρα θα εξεταστούν όλα τα βήματα που πραγματοποιήθηκαν προκειμένου να καθαριστούν τα δεδομένα.

- Αρχικά, εισάγουμε τις απαραίτητες βιβλιοθήκες numpy και pandas και διαβάζουμε το αρχείο με τα LA crimes προκειμένου να το φορτώσουμε στην μνήμη RAM, αποθηκεύοντας το σε ένα dataframe.

```
In [1]: import pandas as pd
import numpy as np
```

- Reading the csv file

	DR_NO	Date Rptd	DATE OCC	TIME OCC	AREA	AREA NAME	Rpt Dist No	Part 1-2	Crm Cd	Crm Cd Desc	...	Status	Status Desc	Crm Cd 1	Crm Cd 2	Crm Cd 3	Crm Cd 4	LOCATION	Cross St
0	1307355	02/20/2010 12:00:00 AM	02/20/2010 12:00:00 AM	1350	13	Newton	1385	2	900	VIOLATION OF COURT ORDER	...	AA	Adult Arrest	900.0	NaN	NaN	NaN	300 E GAGE AV	N
1	11401303	09/13/2010 12:00:00 AM	09/12/2010 12:00:00 AM	45	14	Pacific	1485	2	740	VANDALISM - FELONY (\$400 & OVER, ALL CHURCH VA...	...	IC	Invest Cont	740.0	NaN	NaN	NaN	SEPULVEDA BL	MANCHESTER
2	70309629	08/09/2010 12:00:00 AM	08/09/2010 12:00:00 AM	1515	13	Newton	1324	2	946	MISCELLANEOUS CRIME	...	IC	Invest Cont	946.0	NaN	NaN	NaN	1300 E 21ST ST	N
3	90631215	01/05/2010 12:00:00 AM	01/05/2010 12:00:00 AM	150	6	Hollywood	646	2	900	VIOLATION OF COURT ORDER	...	IC	Invest Cont	900.0	998.0	NaN	NaN	CAHUENGA BL	HOLLYWOOD
4	100100501	01/03/2010 12:00:00 AM	01/02/2010 12:00:00 AM	2100	1	Central	176	1	122	RAPE ATTEMPTED	...	IC	Invest Cont	122.0	NaN	NaN	NaN	8TH ST	SAN PED

- Ελέγχουμε την στήλη DR_NO και βλέπουμε ότι κάθε περιστατικό έχει έναν μοναδικό κωδικό DR_NO

DR_NO column

- We see that the column DR_NO is a unique code for each row, that means each crime reported has a different code

```
In [3]: la_crimes['DR_NO'].value_counts()

Out[3]: 151001085    1
        161811979    1
        111721424    1
        121320908    1
        121316814    1
        ...
        130712581    1
        130718726    1
        151714826    1
        151712779    1
        151001091    1
Name: DR_NO, Length: 2114699, dtype: int64
```

➤ Ακολουθούν οι στήλες Date Rptd και DATE OCC.

Βλέπουμε ότι και για τις δύο στήλες, όλες οι ώρες των περιστατικών έχουν οριστεί ως 00:00 AM. Έτσι, δεν προσφέρουν περαιτέρω πληροφορία και μπορούμε να τις διαγράψουμε για να εξοικονομήσουμε χώρο.

Επίσης αλλάζουμε το format των ημερομηνιών από MM/DD/YYYY σε YYYY-MM-DD, για να τις εισαγάγουμε εύκολα έπειτα μέσα στην αποθήκη δεδομένων.

Date Rptd and DATE OCC columns

- We see that all the dates have the time of 12:00:00 AM

```
In [4]: la_crimes['Date Rptd'].str[-8:].value_counts() #selecting only the time and getting the different values of it

Out[4]: 00:00 AM    2114699
Name: Date Rptd, dtype: int64

In [5]: la_crimes['DATE OCC'].str[-8:].value_counts() #selecting only the time and getting the different values of it

Out[5]: 00:00 AM    2114699
Name: DATE OCC, dtype: int64
```

- As result, we can delete the data of the time because it does not give additional information to us

```
In [6]: la_crimes['Date Rptd'] = la_crimes['Date Rptd'].str[:10] #only keeping the date of the date without the time

In [7]: la_crimes['DATE OCC'] = la_crimes['DATE OCC'].str[:10] #only keeping the date
```

- Now we are going to bring the date format to the appropriate one for databases: YYYY-MM-DD

```
In [8]: la_crimes['Date Rptd'] = pd.to_datetime(la_crimes['Date Rptd'])
la_crimes['DATE OCC'] = pd.to_datetime(la_crimes['DATE OCC'])
la_crimes['Date Rptd'].sample(5)

Out[8]: 599500    2012-05-08
       677507    2013-07-15
       1360371   2016-03-21
       1620965   2017-06-05
       1847089   2018-01-23
Name: Date Rptd, dtype: datetime64[ns]
```

➤ Ακολουθεί η στήλη TIME OCC

Η στήλη αυτή βρίσκεται στη μορφή HHMM. Αυτό δεν μας βολεύει και έτσι αλλάζουμε το format σε HH:MM

Επίσης ενώνουμε τις στήλες DATE OCC και TIME OCC έτσι ώστε όλη η χρονική πληροφορία του περιστατικού να περιέχεται σε μια στήλη μόνο. Με αυτό τον τρόπο εξοικονομείται χώρος και δεν επαναλαμβάνουμε πληροφορίες. Μάλιστα, το νέο format της στήλης DATE OCC είναι YYYY-MM-DD hh:mm:ss. Τέλος διαγράφουμε την στήλη TIME OCC

TIME OCC column

- Now we have to fix the column `TIME OCC` to a '24hour:minutes' format
- First we transform the column to datatype string because it was integer
- Then we add '00' in front of every date (because the rows with time occurred at 12AM have no values for hours)
- Lastly we only select the right string characters

```
In [9]: la_crimes['TIME OCC'] = la_crimes['TIME OCC'].astype(str)

In [10]: la_crimes['TIME OCC'] = '00'+la_crimes['TIME OCC']

In [11]: la_crimes['TIME OCC'] = la_crimes['TIME OCC'].str[-4:-2] + ':' + la_crimes['TIME OCC'].str[-2:]

      • We join the columns DATE OCC and TIME OCC into the column DATE OCC so that it will be in the format YYYY-MM-DD hh:mm. It will be saved as datatype datetime YYYY-MM-DD hh:mm:ss in the database
      • We drop the column TIME OCC

In [12]: la_crimes['DATE OCC'] = la_crimes['DATE OCC'].astype(str) + ' ' + la_crimes['TIME OCC']
la_crimes['DATE OCC'] = pd.to_datetime(la_crimes['DATE OCC'])

In [13]: la_crimes = la_crimes.drop(['TIME OCC'], axis=1)
```

➤ Τώρα εξετάζεται η στήλη Mocodes

Η συγκεκριμένη στήλη βλέπουμε ότι περιέχει κάποιες null τιμές. Αντικαθιστούμε τις null τιμές με την τιμή 'unknown'

Mocodes column

We see that some crime reports have null values for the column of `Mocodes`

- We replace those values with the value `unknown`

```
In [14]: la_crimes.loc[la_crimes['Mocodes'].isna()]

Out[14]:
   DR_NO Date Rptd  DATE OCC AREA AREA NAME Rpt Dist No Part 1-2 Crm Cd Crm Cd Desc Mocodes ... Status Status Desc Crm Cd 1 Crm Cd 2 Crm Cd 3 Crm Cd 4 LOCATI
15 100100535 2010-01-17 2010-01-16 1 Central 185 2 946 MISCELLANEOUS OTHER CRIME NaN ... IC Invest Cont 946.0 999.0 NaN NaN NaN 30 OLYMPIC
28 100100578 2010-02-05 2010-02-03 1 Central 185 2 946 MISCELLANEOUS OTHER CRIME NaN ... IC Invest Cont 946.0 999.0 NaN NaN NaN 1200 MAP
51 100100654 2010-02-27 2010-02-27 1 Central 174 2 946 MISCELLANEOUS OTHER CRIME NaN ... AA Adult Arrest 946.0 NaN NaN NaN NaN W 7TH
79 100100730 2010-03-23 2010-03-20 1 Central 111 2 647 THROWING OBJECT AT MOVING VEHICLE NaN ... IC Invest Cont 647.0 NaN NaN NaN CESAF CHAV
102 100100786 2010-04-08 2010-04-08 1 Central 161 1 510 VEHICLE - STOLEN NaN ... IC Invest Cont 510.0 520.0 NaN NaN FRANCISI

In [15]: indexes_null_mo = la_crimes.loc[la_crimes ['Mocodes'].isna()].index
la_crimes.loc[indexes_null_mo, 'Mocodes'] = 'unknown'
```

➤ Στήλη Vict Age

Εξετάζοντας διαφορετικές τιμές τις ηλικίας, παρατηρούμε ότι κάποιες ηλικίες έχουν αρνητικές τιμές που δεν έχει νόημα. Αντικαθιστούμε τις αρνητικές ηλικίες με την ηλικία 0 που θεωρούμε και missing value.

Vict Age column

- Now we are going to fix the column `Vict Age` that don't make any sense
- Specifically we are going to change all negative age numbers to `0` which we consider missing value.

```
In [16]: la_crimes['Vict Age'].value_counts()
Out[16]: 0      369886
25     48101
26     47469
27     47011
24     46739
...
-7      15
-8      7
-9      4
114     1
118     1
Name: Vict Age, Length: 110, dtype: int64

In [17]: indexes_neg_age = la_crimes.loc[la_crimes['Vict Age'] < 0]['Vict Age'].index #finding the indexes of negative ages
la_crimes.loc[indexes_neg_age, 'Vict Age'] = 0
```

➤ Στήλη Vict Sex

Σύμφωνα με την περιγραφή των δεδομένων για την στήλη Vict Sex έχουμε μόνο τις τιμές M (male), F (female) και X (unknown).

Παρατηρώντας τις διαφορετικές τιμές της στήλης, βλέπουμε και μερικές άλλες τιμές όπως H και N, τις οποίες και αλλάζουμε σε X (unknown) (θεωρώντας ότι έχει γίνει λάθος στην εισαγωγή των δεδομένων)

Vict Sex column

- Now we fix the column `Vict Sex` which is the victim's sex
- According to the description the values are: F - Female, M - Male, X - Unknown

```
In [18]: la_crimes['Vict Sex'].value_counts()
Out[18]: M      974309
F      888499
X      55129
H       73
N       17
-        1
Name: Vict Sex, dtype: int64

• We see there are a bunch of irrelevant values which we change to 'X'

In [19]: indexes_wrong_sex = la_crimes.loc[(la_crimes['Vict Sex'] != 'M') & (la_crimes['Vict Sex'] != 'F')].index
la_crimes.loc[indexes_wrong_sex, 'Vict Sex'] = 'X'
```

➤ Στήλη Vict Descent

Βλέπουμε ότι αρκετά περιστατικά έχουν null τιμές ή την τιμή '-' για την συγκεκριμένη στήλη. Αντικαθιστούμε αυτές τις τιμές με την τιμή 'X' που έχει οριστεί ως unknown victim descent.

```
In [20]: print("The reported crimes with Nan (null) values for the column Vict Descent are:")
len(la_crimes.loc[la_crimes['Vict Descent'].isna()])

The reported crimes with Nan (null) values for the column Vict Descent are:
Out[20]: 196718

In [21]: la_crimes['Vict Descent'].value_counts()

Out[21]: H    725348
W    510158
B    335102
O    202969
X    78147
A    51109
K    8141
F    2553
C    1061
I    945
J    418
P    343
V    201
U    190
Z    136
G    85
S    31
D    23
L    18
-    3
Name: Vict Descent, dtype: int64

In [22]: indexes_wrong_descent = la_crimes.loc[(la_crimes['Vict Descent'] == '-') | (la_crimes['Vict Descent'].isna())].index
la_crimes.loc[indexes_wrong_descent,'Vict Descent'] = 'X'
```

➤ Στήλη Premis Cd & Premis Desc

Βλέπουμε ότι αρκετά περιστατικά έχουν null τιμές για την στήλη Premis Cd. Αντικαθιστούμε αυτές τις null τιμές με την τιμή 0 που θεωρούμε ως missing value

Premis Cd and Premis Desc columns

Now we see that many rows have null values for the column Premis Cd

- Because this column has float values, we replace the null values with 0

```
In [23]: la_crimes.loc[la_crimes['Premis Cd'].isna()]

Out[23]:
   DR_NO Date Rptd DATE OCC AREA AREA NAME Rpt Dist No Part 1-2 Crm Cd Crm Cd Desc Mocodes ... Status Status Desc Crm Cd 1 Crm Cd 2 Crm Cd 3 Crm Cd 4 LO
   6590 100121447 2010- 12-12 2010-12-11:50:00 Central 185 1 110 CRIMINAL HOMICIDE unknown ... AA Adult Arrest 110.0 NaN NaN NaN NaN OLY
   32148 100913648 2010- 06-21 2010-06-20 14:35:00 Van Nuys 915 1 510 VEHICLE - STOLEN unknown ... IC Invest Cont 510.0 NaN NaN NaN NaN 7
   67340 100816222 2010- 09-03 2010-04-16 00:01:00 West LA 803 2 813 CHILD ANNOYING (17YRS & UNDER) unknown ... IC Invest Cont 813.0 NaN NaN NaN NaN MAND
   68276 100818222 2010- 11-18 2010-10-16 16:00:00 West LA 811 2 812 CRM AGNST CHLD (13 OR UNDER)(14-15 & SUSP 10 ... unknown ... IC Invest Cont 812.0 NaN NaN NaN NaN PUI

In [24]: la_crimes.loc[la_crimes['Premis Desc'] == 'unknown'][['Premis Cd','Premis Desc']]

Out[24]:
Premis Cd Premis Desc
```

```
In [25]: indexes_null_premiscd = la_crimes.loc[la_crimes ['Premis Cd'].isna()].index
la_crimes.loc[indexes_null_premiscd,'Premis Cd'] = 0
```

Επίσης, αρκετά περιστατικά έχουν null τιμές για την στήλη Premis Desc. Αντικαθιστούμε αυτές τις null τιμές με την τιμή ‘unknown’ που θεωρούμε ως missing value

Now we see that many rows have null values for the column Premis Desc

- We replace those null values with unknown

In [26]: la_crimes.loc[la_crimes['Premis Desc'].isna()]

Out[26]:

	DR_NO	Date Rptd	DATE OCC	AREA	AREA NAME	Rpt Dist No	Part 1-2	Crm Cd	Crm Cd Desc	Mocodes	...	Status	Status Desc	Crm Cd 1	Crm Cd 2	Crm Cd 3	Crm Cd 4	LOCATION
6590	100121447	2010-12-12	2010-12-12 11:50:00	1	Central	185	1	110	CRIMINAL HOMICIDE	unknown	...	AA	Adult Arrest	110.0	NaN	NaN	NaN	2010 OLYMPIC
32148	100913648	2010-06-21	2010-06-20 14:35:00	9	Van Nuys	915	1	510	VEHICLE - STOLEN	unknown	...	IC	Invest Cont	510.0	NaN	NaN	NaN	7000 VAN NUYS
67340	100816222	2010-09-03	2010-04-16 00:01:00	8	West LA	803	2	813	CHILD ANNOYING (17YRS & UNDER)	unknown	...	IC	Invest Cont	813.0	NaN	NaN	NaN	2010 MANDAVILLE
68276	100818222	2010-11-18	2010-10-16 16:00:00	8	West LA	811	2	812	CRM AGNST CHLD (13 OR UNDER) (14-15 & SUSP 10 ...	unknown	...	IC	Invest Cont	812.0	NaN	NaN	NaN	11000 PULGAS

In [27]: indexes_null_premisdesc = la_crimes.loc[la_crimes ['Premis Desc'].isna()].index
la_crimes.loc[indexes_null_premisdesc,'Premis Desc'] = 'unknown'

Θεωρητικά, όλα τα περιστατικά με τιμή ‘unknown’ στην στήλη Premis Desc, πρέπει να έχουν την τιμή 0 για την στήλη Premis Cd. Αυτό όμως δεν συμβαίνει για κάποια περιστατικά

Theoretically, for the rows with Premis Desc = 'unknown', the column Premis Cd should be equal to 0 (unknown). However, there are some rows with Premis Cd different than 0

In [28]: la_crimes.loc[la_crimes['Premis Desc']=='unknown'][['Premis Desc','Premis Cd']]

Out[28]:

Premis Desc	Premis Cd
6590	unknown
32148	unknown
67340	unknown
68276	unknown
71523	unknown
...	...
2109009	418.0
2109581	256.0
2113695	256.0
2114205	256.0
2114388	418.0

188 rows × 2 columns

In detail, the rows with Premis Desc = 'unknown', have the values of 0 or 256 or 418 or 838

In [29]: la_crimes.loc[la_crimes['Premis Desc']=='unknown']['Premis Cd'].unique()

Out[29]: array([0., 838., 418., 256.])

Μάλιστα, μετά από ελέγχους φαίνεται ότι τα περιστατικά με τις τιμές 256, 418 και 838 για την στήλη Premis Cd έχουν τιμή ‘unknowm’ στην στήλη Premis Desc. Προκειμένου να υπάρχει μια δομή στα δεδομένα, αλλάζουμε τις τιμές 256, 418, 838 σε 0 για την στήλη Premis Cd. Έτσι, όλα τα περιστατικά με ‘unknowm’ Premis Desc έχουν τιμή 0 για την στήλη Premis Cd και το αντίστροφο.

In detail, the rows with `Premis Desc` = 'unknown', have the values of 0 or 256 or 418 or 838

```
In [29]: la_crimes.loc[la_crimes['Premis Desc']=='unknown']['Premis Cd'].unique()
Out[29]: array([ 0., 838., 418., 256.])
```

- Next up we see that all the rows with `Premis Cd` = 418, have `Premis Desc` = 'unknown'
- Thus we can change `Premis Cd` to 0 (unknown)

```
In [30]: cd418 = la_crimes.loc[la_crimes['Premis Cd']==418][['Premis Desc','Premis Cd']]
cd418['Premis Desc'].unique()
Out[30]: array(['unknown'], dtype=object)
```

- Next up we see that all the rows with `Premis Cd` = 256, have `Premis Desc` = 'unknown'
- Thus we can change `Premis Cd` to 0 (unknown)

```
In [31]: cd256 = la_crimes.loc[la_crimes['Premis Cd']== 256][['Premis Desc','Premis Cd']]
cd256['Premis Desc'].unique()
Out[31]: array(['unknown'], dtype=object)
```

- Next up we see that all the rows with `Premis Cd` = 838, have `Premis Desc` = 'unknown'
- Thus we can change `Premis Cd` to 0 (unknown)

```
In [32]: cd838 = la_crimes.loc[la_crimes['Premis Cd']== 838][['Premis Desc','Premis Cd']]
cd838['Premis Desc'].unique()
Out[32]: array(['unknown'], dtype=object)
```

- We change all the rows with `Premis Cd` equal to 256, 418 or 838, to 0 (unknown)

```
In [33]: indexes_wrong_premiscd = la_crimes.loc[(la_crimes ['Premis Cd'] == 256) |
                                             (la_crimes ['Premis Cd'] == 418) |
                                             (la_crimes ['Premis Cd'] == 838)].index
la_crimes.loc[indexes_wrong_premiscd,'Premis Cd'] = 0
```

- Lastly, we transform `Premis Cd` from float to integer

```
In [34]: la_crimes['Premis Cd'] = la_crimes['Premis Cd'].astype(int)
```

➤ Στήλες Weapon Used Cd & Weapon Desc

Βλέπουμε τώρα ότι αρκετά περιστατικά έχουν null τιμές για την στήλη `Weapon Used Cd`. Αντικαθιστούμε αυτές τις null τιμές με την τιμή 0 που θεωρούμε ως missing value.

Weapon Used Cd and Weapon Desc columns

Some rows have null values for the column `Weapon Used Cd`

- We replace those null values with 0 because this column contains float datatypes

	DR_NO	Date Rptd	DATE OCC	AREA	AREA NAME	Rpt Dist No	Part 1-2	Crm Cd	Crm Cd Desc	Mocodes	...	Status	Status Desc	Crm Cd 1	Crm Cd 2	Crm Cd 3	Crm Cd 4	LOCAT
0	1307355	2010-02-20	2010-02-20 13:50:00	13	Newton	1385	2	900	VIOLATION OF COURT ORDER	0913 1814 2000	...	AA	Adult Arrest	900.0	NaN	NaN	NaN	300 E G
1	11401303	2010-09-13	2010-09-12 00:45:00	14	Pacific	1485	2	740	VANDALISM - FELONY (\$400 & OVER, ALL CHURCH VA...	0329	...	IC	Invest Cont	740.0	NaN	NaN	NaN	SEPULV
2	70309629	2010-08-09	2010-08-09 15:15:00	13	Newton	1324	2	946	OTHER MISCELLANEOUS CRIME	0344	...	IC	Invest Cont	946.0	NaN	NaN	NaN	1300 E 2
5	100100506	2010-01-05	2010-01-04 16:50:00	1	Central	162	1	442	SHOPLIFTING - PETTY THEFT (\$950 & UNDER)	0344 1402	...	AA	Adult Arrest	442.0	NaN	NaN	NaN	700 W 7TH
6	100100508	2010-01-08	2010-01-07 20:05:00	1	Central	182	1	330	BURGLARY FROM VEHICLE	0344	...	IC	Invest Cont	330.0	NaN	NaN	NaN	PICO

```
In [36]: indexes_null_weaponcd = la_crimes.loc[la_crimes ['Weapon Used Cd'].isna()].index
la_crimes.loc[indexes_null_weaponcd,'Weapon Used Cd'] = 0
```

Επίσης, κάποια περιστατικά έχουν null τιμές για την στήλη Weapon Desc. Αντικαθιστούμε αυτές τις null τιμές με την τιμή 'unknown' που θεωρούμε ως missing value

Ακόμα, μετατρέπουμε την στήλη Weapon Used Cd από float σε integer για να μην αποθηκεύσουμε τα '.00' στη βάση που θα δημιουργούσαν προβλήματα έπειτα.

Also rows have null values for the column Weapon Desc

- We replace those null values with unknown because this column contains object datatypes

	DR_NO	Date Rptd	DATE OCC	AREA	AREA NAME	Rpt Dist No	Part 1,2	Crm Cd	Crm Cd Desc	Mocodes	...	Status	Status Desc	Crm Cd 1	Crm Cd 2	Crm Cd 3	Crm Cd 4	LOCATI
0	1307355	2010-02-20	2010-13:50:00	13	Newton	1385	2	900	VIOLATION OF COURT ORDER	0913 1814 2000	...	AA	Adult Arrest	900.0	NaN	NaN	NaN	300 E G
1	11401303	2010-09-13	2010-09-12 00:45:00	14	Pacific	1485	2	740	VANDALISM - FELONY (\$400 & OVER, ALL CHURCH VA...)	0329	...	IC	Invest Cont	740.0	NaN	NaN	NaN	SEPULV
2	70309629	2010-08-09	2010-08-09 15:15:00	13	Newton	1324	2	946	OTHER MISCELLANEOUS CRIME	0344	...	IC	Invest Cont	946.0	NaN	NaN	NaN	1300 E 2
5	100100506	2010-01-05	2010-01-04 16:50:00	1	Central	162	1	442	SHOPLIFTING - PETTY THEFT (\$950 & UNDER)	0344 1402	...	AA	Adult Arrest	442.0	NaN	NaN	NaN	700 W 7TH
6	100100508	2010-01-08	2010-01-07 20:25:00	1	Central	182	1	330	BURGLARY FROM VEHICLE	0344	...	IC	Invest Cont	330.0	NaN	NaN	NaN	PIC

```
In [38]: indexes_null_weapondesc = la_crimes.loc[la_crimes['Weapon Desc'].isna()].index
la_crimes.loc[indexes_null_weapondesc,'Weapon Desc'] = 'unknown'
```

Lastly, we transform the column Weapon Used Cd from float to integer

```
In [39]: la_crimes['Weapon Used Cd'] = la_crimes['Weapon Used Cd'].astype(int)
```

➤ Στήλες Status and Status Desc

Παρατηρούμε ότι υπάρχουν κάποιες διαφορές στις στήλες Status and Status Desc. Θεωρητικά, αφού η κάθε τιμή της μιας στήλης αντιστοιχεί με μια συγκεκριμένη τιμή της άλλης, ο αριθμός εμφάνισης κάθε τιμής θα έπρεπε να ο ίδιος στις δύο στήλες.

Status and Status Desc columns

- Now we check the status of the crime incident
- We see that the columns `Status` and `Status Desc` have some differences.

```
In [40]: la_crimes['Status'].value_counts()
```

```
Out[40]: IC    1623829
AO    250589
AA    219081
JA    15864
JO    5301
CC     29
19      1
TH      1
13      1
Name: Status, dtype: int64
```

```
In [41]: la_crimes['Status Desc'].value_counts()
```

```
Out[41]: Invest Cont    1623829
Adult Other    250589
Adult Arrest    219081
Juv Arrest    15864
Juv Other    5301
UNK        35
Name: Status Desc, dtype: int64
```

Παρατηρούμε ότι όλα τα περιστατικά με Status CC, 19, TH και 13 έχουν για την στήλη Status Desc την τιμή 'UNK'. Για αυτό αλλάζουμε τις τιμές 19, TH, 13 ή null σε CC που θεωρείται και τιμή missing value.

Επιπρόσθετα, κάποια περιστατικά έχουν null τιμές για την στήλη Status Desc. Αντικαθιστούμε αυτές τις null τιμές με την τιμή 'UNK' (unknown) που θεωρούμε ως missing value

- We observe that all the values with a Status CC , 19 , TH , 13 have a Status Description UNK
- That's why we change the rows with a Status 19 , TH , 13 (because those only appear once) or null to CC

```
In [42]: la_crimes.loc[la_crimes['Status Desc'] == 'UNK']
```

	DR_NO	Date Rptd	DATE OCC	AREA	AREA NAME	Rpt Dist No	Part 1-2	Crm Cd	Crm Cd Desc	Mocodes	...	Status	Status Desc	Crm Cd 1	Crm Cd 2	Crm Cd 3	C
100040	101208618	2010-03-02	2010-03:30:00	12	77th Street	1248	1	210	ROBBERY	0305 0342 0344 0416 1008	...	Nan	UNK	210.0	NaN	NaN	N
151803	101700682	2010-03-09	2010-11:55:00	17	Devonshire	1764	2	653	CREDIT CARDS, FRAUD USE (\$950.01 & OVER)	0377 0930 1402 1822	...	CC	UNK	653.0	998.0	NaN	N
160732	101721148	2010-11-15	2010-11-17:00:00	17	Devonshire	1756	2	900	VIOLATION OF COURT ORDER	1501	...	CC	UNK	900.0	NaN	NaN	N
219776	112109831	2011-04-29	2011-04-28 22:00:00	21	Topanga	2139	1	331	THEFT FROM MOTOR VEHICLE - GRAND (\$400 AND OVER)	0344 1202	...	CC	UNK	331.0	NaN	NaN	N

```
In [43]: indexes_status = la_crimes.loc[(la_crimes['Status'] == '19') |
                                         (la_crimes['Status'] == '13') | (la_crimes['Status'] == 'TH') |
                                         (la_crimes['Status'].isna())]
                                         ]['Status'].index #finding the indexes of nan Status or status 19,13,TH
la_crimes.loc[indexes_status, 'Status'] = 'CC'
```

- We consider that the value UNK for the column Status Desc means unknown
- Then we will change the Nan values for column Status Desc to UNK

```
In [44]: indexes_status_desc = la_crimes.loc[la_crimes['Status Desc'].isna()]['Status'].index #finding the indexes of nan Status
la_crimes.loc[indexes_status_desc, 'Status Desc'] = 'UNK'
```

➤ Στήλες Crm Cd 1, 2, 3, 4

Σύμφωνα με την περιγραφή των στηλών Crm Cd 1,2,3,4, η στήλη Crm Cd 1 αφορά το πρωταρχικό και πιο σοβαρό έγκλημα και οι υπόλοιπες 3 αφορούν λιγότερο σοβαρές παραβάσεις. Με βάση αυτές τις περιγραφές καταλήγουμε στα συμπεράσματα ότι δεν έχει νόημα η ύπαρξη περιστατικού με τιμή null για το Crm Cd 1 και τιμή μη-null για τη στήλη Crm Cd 2. Επίσης, σε κάθε περίπτωση περιστατικού πρέπει πρώτα να γεμίζουν (παίρνει μη null τιμές) η στήλη Crm Cd 1 μετά η στήλη Crm Cd 2 ύστερα Crm Cd 3 και τέλος η στήλη Crm Cd 4. Αυτά ακριβώς τα συμπεράσματα εφαρμόζονται εδώ.

Crm Cd columns

Also, reading the documentation of the columns Crm Cd indicates the crime committed. Crm Cd 1 is the primary and most serious one. Crime Code 2, 3, and 4 are respectively less serious offenses

- As a result it doesn't make sense to only have a Crm Cd 2 offense with no Crm Cd 1 offense
- Thus we will update all crime reports with null Crm Cd 1 and some Crm Cd 2 or Crm Cd 3 offenses so that in every case Crm Cd 1 has a value.
- If there were 2 offenses then there will be only values for Crm Cd 1 and Crm Cd 2 and so on

```
In [45]: indexes_crmlnull = la_crimes.loc[(la_crimes ['Crm Cd 1'].isna()) ].index
la_crimes.loc[indexes_crmlnull,:]
```

Out[45]:

DR_NO	Date Rptd	DATE OCC	AREA	AREA NAME	Rpt Dist No	Part 1-2	Crm Cd	Crm Cd Desc	Mocodes	...	Status	Status Desc	Crm Cd 1	Crm Cd 2	Crm Cd 3	Crm Cd 4	LOCATION
55532	100707214	2010-03-14 02:30:00	2010-03-13	7 Wilshire	767	2	624	BATTERY - SIMPLE ASSAULT	0344 0400 0416 1300	...	IC	Invest Cont	NaN	624.0	NaN	NaN	PICO BL
288235	110310134	2011-04-08 08:00:00	2011-03-26	3 Southwest	312	2	942	BRIBERY	1300 1402	...	IC	Invest Cont	NaN	942.0	99.0	NaN	ROSELAND ST
358506	110811926	2011-07-01 20:11:00	2011-07-01	8 West LA	835	1	210	ROBBERY	unknown	...	IC	Invest Cont	NaN	210.0	NaN	NaN	11000 SANTA MONICA BL
507665	120325216	2012-11-19 19:30:00	2012-11-19	3 Southwest	329	1	440	THEFT PLAIN - PETTY (\$950 & UNDER)	0344 0913 0906 0224	...	IC	Invest Cont	NaN	440.0	NaN	NaN	500 W 27TH ST

```
In [46]: la_crimes.loc[indexes_crmlnull,'Crm Cd 1'] = la_crimes.loc[indexes_crmlnull,'Crm Cd 2']
la_crimes.loc[indexes_crmlnull,'Crm Cd 2'] = la_crimes.loc[indexes_crmlnull,'Crm Cd 3']
la_crimes.loc[indexes_crmlnull,'Crm Cd 3'] = la_crimes.loc[indexes_crmlnull,'Crm Cd 4']
```

Σε αυτό το σημείο αντικαθιστούμε τις null τιμές στις στήλες Crm Cd 2, 3, 4 με την τιμή -1 που θεωρούμε ως no extra crime committed. Επίσης, μετατρέπουμε τις στήλες Crm Cd 1,2, 3, 4 από float σε int, για να μην αποθηκεύσουμε τα '.00' στο τελικό αρχείο, πράγμα που θα δημιουργούσε προβλήματα στην αποθήκη δεδομένων.

We fill up all the null values for `Crm Cd 2` and `Crm Cd 3` and `Crm Cd 4` with -1 (meaning none is found) because those columns consist of float datatype

```
In [47]: la_crimes[['Crm Cd 2']] = la_crimes[['Crm Cd 2']].fillna(-1)
la_crimes[['Crm Cd 3']] = la_crimes[['Crm Cd 3']].fillna(-1)
la_crimes[['Crm Cd 4']] = la_crimes[['Crm Cd 4']].fillna(-1)
```

Lastly, we transform the columns `Crm Cd 1`, `Crm Cd 2`, `Crm Cd 3`, `Crm Cd 4` from float to integer

```
In [48]: la_crimes['Crm Cd 1'] = la_crimes['Crm Cd 1'].astype(int)
la_crimes['Crm Cd 2'] = la_crimes['Crm Cd 2'].astype(int)
la_crimes['Crm Cd 3'] = la_crimes['Crm Cd 3'].astype(int)
la_crimes['Crm Cd 4'] = la_crimes['Crm Cd 4'].astype(int)
```

Σε αυτό το σημείο γίνεται έλεγχος, ώστε να είμαστε σίγουροι ότι οι στήλες `Crm Cd` και `Crm Cd 1` έχουν τις ίδιες τιμές ως τα πιο σοβαρά εγκλήματα. Έτσι, στα περιστατικά που δεν συμβαίνει αυτό, το φτιάχνουμε. Συγκεκριμένα, πρώτα παίρνουμε την περίπτωση που οι στήλες `Crm Cd` και `Crm Cd 2` έχουν την ίδια τιμή και έπειτα την περίπτωση που οι στήλες `Crm Cd` και `Crm Cd 3` έχουν την ίδια τιμή (για την περίπτωση που οι στήλες `Crm Cd` και `Crm Cd 4` είναι ίδιες, δεν υπάρχει κάποιο περιστατικό). Έτσι αλλάζουμε τις τιμές των στηλών `Crm Cd 2` και `Crm Cd 1` ή `Crm Cd 3` και `Crm Cd 1`, ώστε σε κάθε περίπτωση οι στήλες `Crm Cd` και `Crm Cd 1` να έχουν την ίδια ακριβώς τιμή. Με αυτό τον τρόπο, μπορούμε τώρα να διαγράψουμε την στήλη `Crm Cd 1` αφού δεν μας προσφέρει περεταίρω πληροφορία.

- We check if for every crime incident, the column `Crm Cd` and `Crm Cd 1` are the same.
- Some are not the same so we consider them mistaken because `Crm Cd 1` is the primary crime committed and `Crm Cd` describes the crime committed
- For those incidents we set `Crm Cd 1` to be equal to `Crm Cd`
- First case is that `Crm Cd` has the same value as `Crm Cd 2`
- Then we just have to swap `Crm Cd 1` and `Crm Cd 2`

```
In [49]: indexes_crm_crm2 = la_crimes.loc[la_crimes['Crm Cd'] == la_crimes[['Crm Cd 2']].index]
la_crimes.loc[indexes_crm_crm2, 'Crm Cd 2'] = la_crimes.loc[indexes_crm_crm2, 'Crm Cd 1'].copy()
la_crimes.loc[indexes_crm_crm2, 'Crm Cd 1'] = la_crimes.loc[indexes_crm_crm2, 'Crm Cd'].copy()
```

- Second case is that `Crm Cd` has the same value as `Crm Cd 3`
- Then we just have to swap `Crm Cd 1` and `Crm Cd 3`

```
In [50]: indexes_crm_crm3 = la_crimes.loc[la_crimes['Crm Cd'] == la_crimes[['Crm Cd 3']].index]
la_crimes.loc[indexes_crm_crm3, 'Crm Cd 3'] = la_crimes.loc[indexes_crm_crm3, 'Crm Cd 1'].copy()
la_crimes.loc[indexes_crm_crm3, 'Crm Cd 1'] = la_crimes.loc[indexes_crm_crm3, 'Crm Cd'].copy()
```

- Third case is that `Crm Cd` has the same value as `Crm Cd 4`
- However, there is no incident like that

```
In [51]: la_crimes.loc[la_crimes['Crm Cd'] == la_crimes[['Crm Cd 4']].index]
```

```
Out[51]: Int64Index([], dtype='int64')
```

- Lastly, because `Crm Cd` and `Crm Cd 1` are the same now, we can drop `Crm Cd 1`

```
In [52]: la_crimes = la_crimes.drop(['Crm Cd 1'], axis=1)
```

➤ Στήλη LOCATION

Για την συγκεκριμένη στήλη φαίνεται ότι διάφορα περιστατικά έχουν πάρα πολλά κενά μεταξύ των λέξεων. Αυτό δεν μας αρέσει αφού δεσμεύει περισσότερο χώρο και σε περίπτωση ομαδοποίησης με βάση την τιμή του location μπορεί να πάρουμε λανθασμένα αποτελέσματα (π.χ. '9300 TAMPA AV', '9300 TAMPA AV' δεν θα ομαδοποιούντουσαν μαζί).

LOCATION Column

- Some locations have a bunch of white spaces in between their words

```
In [53]: la_crimes['LOCATION'].value_counts()

Out[53]: 6TH                      ST      4756
          7TH                      ST      3774
          9300  TAMPA                  AV      3658
          6TH                      BL      3235
          6600  TOPANGA CANYON        BL      3064
                                         ...
          LEADWELL                   AV      1
          2400  HOOVER                 ST      1
          18100 W ANDREA CIRCL       BL      1
          5200  HERITAGE                AV      1
          INDIANA                   CT      1
Name: LOCATION, Length: 75251, dtype: int64
```

- The unnecessary white spaces are removed from the middle, front and back of the words

```
In [54]: la_crimes['LOCATION'] = la_crimes['LOCATION'].str.replace(' +', ' ').str.strip()
```

➤ Στήλη Cross Street

Παρατηρούμε ότι για την συγκεκριμένη στήλη οι περισσότερες τιμές είναι null. Συγκεκριμένα, οι 1,76 εκ. από τις 2,1 εκ. (84%) είναι null. Έτσι αποφασίζουμε αφού έχουμε ήδη την στήλη location, να διαγράψουμε την συγκεκριμένη στήλη επειδή παρέχει πολύ περιορισμένη πληροφόρηση.

- Now we observe that for most crime incidents this column has the value unknown
- In detail that is true, for the 1.76 million incidents out of the 2.1 million incidents that there are
- Thus we decide to drop this column

In [58]: la_crimes.loc[la_crimes['Cross Street'] == 'unknown']																			
2114693	190506304	2019-02-22	2019-02-22	5	Harbor	569	2	627	CHILD ABUSE (PHYSICAL) - SIMPLE ASSAULT	0443 0419 0416 1259	...	STRONG-ARM (HANDS, FIST, FEET OR BODILY FORCE)	AO	Adult Other	-1	-1	-1		
2114694	190608903	2019-03-28	2019-03-28	6	Hollywood	644	1	648	ARSON	0601 1501	...	FIRE	IC	Invest Cont	-1	-1	-1		
2114697	191716777	2019-10-17	2019-10-16	17	Devonshire	1795	1	420	THEFT FROM MOTOR VEHICLE - PETTY (\$550 & UNDER)	unknown	...	unknown	IC	Invest Cont	-1	-1	-1		
2114698	190805435	2019-02-01	2019-02-01	8	West LA	852	1	330	BURGLARY FROM VEHICLE	1302 1609 0358 1307 0344 0377 0321	...	unknown	IC	Invest Cont	-1	-1	-1		
1759334 rows × 26 columns																			

```
In [59]: la_crimes = la_crimes.drop(['Cross Street'], axis=1)
```

➤ Στήλες LONGTITUDE & LATITUDE

Βλέπουμε ότι αρκετά περιστατικά έχουν την τιμή 0 για τις στήλες LONGTITUDE και LATITUDE. Το 0° γεωγραφικό πλάτος και μήκος αντιστοιχεί κάπου στον κόλπο της Γουινέας, με αποτέλεσμα να θεωρούμε την συγκεκριμένη τιμή ως missing value

LONGTITUDE-LATITUDE

- We see that some crimes have values for longitude and latitude equal to 0
- Those values are considered missing because they correspond to the Gulf of Guinea.

In [60]: la_crimes.loc[(la_crimes['LON'] == 0) & (la_crimes['LAT'] == 0)]																
Out[60]:																
DR_NO	Date Rptd	DATE OCC	AREA	AREA NAME	Rpt Dist No	Part 1-2	Crm Cd	Crm Cd Desc	Mocodes	...	Weapon Used Cd	Weapon Desc	Status	Status Desc	Crm Cd 2	Crm Cd 3
49703	100618355	2010-07-14 19:00:00	6	Hollywood	665	1	330	BURGLARY FROM VEHICLE	0344 1300 ...	0	unknown	IC	Invest Cont	-1	-1	
49800	100618603	2010-07-19 23:45:00	6	Hollywood	665	2	740	VLANDALISM - FELONY (\$400 & OVER, ALL CHURCH VA...	0329 0906 ...	0	unknown	AA	Adult Arrest	998	-1	
60848	100718479	2010-11-29 16:30:00	7	Wilshire	709	1	230	ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT	0416 1402 ...	122	HECKLER & KOCH 93 SEMIAUTOMATIC ASSAULT RIFLE	IC	Invest Cont	998	-1	
84978	101016365	2010-08-23 10:00:00	10	West	1000	2	626	INTIMATE PARTNER -	0416 1402 ...	0	unknown	IC	Invest Cont	-1	-1	

Data transforming into proper storage format

Η βάση δεδομένων θα οργανωθεί με την μορφή star schema. Έτσι, πρέπει να μετατρέψουμε τα δεδομένα στις κατάλληλες δομές (αρχεία - tables) για να επιτρέψουν την αναπαράσταση τους σε star schema. Μάλιστα, θα υπάρχουν πολλές διαφορετικές διαστάσεις που θα σχετίζονται με κάθε συγκεκριμένο περιστατικό. Έτσι όλα τα περιστατικά εγκλήματος θα βρίσκονται στο fact table LA crimes. Επιπρόσθετα, οι διαστάσεις (dimension tables) θα είναι η τοποθεσία, η ημερομηνία που αναφέρθηκε, η ημερομηνία που συνέβη, χαρακτηριστικά του ύποπτου, χαρακτηριστικά του θύματος, το έγκλημα που διαπράχθηκε, οι εγκαταστάσεις (δηλαδή το είδος της δομής, του οχήματος ή της τοποθεσίας όπου έγινε το έγκλημα), το όπλο που χρησιμοποιήθηκε, το status και οι γεωγραφικές συντεταγμένες.

Τώρα θα δημιουργήσουμε τα συγκεκριμένα dimension tables:

➤ Location Dimension

Σε αυτό το σημείο δημιουργείται η διάσταση Location. Η συγκεκριμένη διάσταση περιλαμβάνει τις στήλες LOCATION, Rpt Dist No, AREA και AREA NAME. Για την δημιουργία της διάστασης καθαρίζουμε τα επαναλαμβανόμενες τιμές του πίνακα la_crimes.

Location dimension

- We start by creating the dimension of the location which will include the location, the district, the area code and the area name
- Particularly we take the columns LOCATION, Rpt Dist No and AREA NAME and clear the duplicate values
- The results are saved in a new dataframe called location_dimension

```
In [61]: la_crimes[['LOCATION', 'Rpt Dist No', 'AREA', 'AREA NAME']].drop_duplicates()
```

```
Out[61]:
```

	LOCATION	Rpt Dist No	AREA	AREA NAME
0	300 E GAGE AV	1385	13	Newton
1	SEPULVEDA BL	1485	14	Pacific
2	1300 E 21ST ST	1324	13	Newton
3	CAHUENGA BL	646	6	Hollywood
4	8TH ST	176	1	Central
...
2114446	8300 81ST ST	1483	14	Pacific
2114462	4500 ALAMEDA	1367	13	Newton
2114538	WASHINGTON	2069	20	Olympic
2114594	CRYSTAL	1133	11	Northeast
2114634	WESTERN CANYON	1161	11	Northeast

123751 rows × 4 columns

Μάλιστα παρατηρούμε ότι για κάθε Rpt Dist No αντιστοιχεί μόνο ένα AREA. Έτσι αν ξέρουμε το district όπου έγινε ένα έγκλημα, μπορούμε εύκολα να μάθουμε και το Area στο οποίο πραγματοποιήθηκε. Το ίδιο, δεν συμβαίνει όμως και για το LOCATION (διεύθυνση), πιθανότατα λόγω συνονόματων διευθύνσεων ή λανθασμένων καταγραφών.

Παράλληλα, το index του νέου dataframe location_dimension που δημιουργήσαμε, αποθηκεύεται σε νέα στήλη προκειμένου να χρησιμοποιηθεί σαν ξένο κλειδί στην συνέχεια.

- We see that each district number (Rpt Dist No) corresponds to one area name
- That means if we just know the district where a crime happened, we can find out the area where it took place because every district belongs to an area
- However, each location can correspond to different districts and areas because of the likelihood of using the same names

```
In [62]: la_crimes[['Rpt Dist No','AREA NAME']].drop_duplicates()
```

```
Out[62]:
```

	Rpt Dist No	AREA NAME
0	1385	Newton
1	1485	Pacific
2	1324	Newton
3	646	Hollywood
4	176	Central
...
1967201	1086	West Valley
2057309	1686	Foothill
2077942	1784	Devonshire
2089763	1647	Foothill
2095715	693	Hollywood

```
In [63]: location_dimension = la_crimes[['LOCATION','Rpt Dist No','AREA NAME']].drop_duplicates()
location_dimension = location_dimension.reset_index(drop = True)
location_dimension['LOCATION_CD'] = location_dimension.index
location_dimension
```

```
Out[63]:
```

	LOCATION	Rpt Dist No	AREA NAME	LOCATION_CD
0	300 E GAGE AV	1385	Newton	0
1	SEPULVEDA BL	1485	Pacific	1
2	1300 E 21ST ST	1324	Newton	2
3	CAHUENGA BL	646	Hollywood	3
4	8TH ST	176	Central	4
...
123746	8300 81ST ST	1483	Pacific	123746
123747	4500 ALAMEDA	1387	Newton	123747
123748	WASHINGTON	2089	Olympic	123748
123749	CRYSTAL	1133	Northeast	123749
123750	WESTERN CANYON	1161	Northeast	123750

Το index του νέου dataframe location_dimension που δημιουργήσαμε, θα χρησιμοποιηθεί σαν ξένο κλειδί που θα συνδέει τον fact table με το συγκεκριμένο dimension table. Έτσι, πρέπει να δημιουργήσουμε μια στήλη ('LOCATION_CD') στο fact table (la_crimes) που θα περιέχει τιμές αυτού του ξένου κλειδιού. Αυτό ακριβώς γίνεται εδώ με την εντολή merge. Τέλος, αφού το fact table πάρει τις ανάλογες τιμές του ξένου κλειδιού, μπορούμε να διαγράψουμε τις στήλες LOCATION, Rpt Dist No, AREA και AREA NAME

- We will use the index column of the new dataframe `location_dimension` as a foreign key and thus we need to add a new column to `la_crimes` containing for each incident the value of this foreign key
- After we do that, the columns `LOCATION`, `Rpt Dist No`, `AREA` and `AREA NAME` are no longer needed in the dataframe `la_crimes` as we have the mapping from the column `LOCATION_CD` to the dimension table `location_dimension`

```
In [64]: la_crimes = pd.merge(la_crimes,
                           location_dimension,
                           on=['LOCATION','Rpt Dist No','AREA NAME'],
                           how='inner')
la_crimes
```

DATE OCC	AREA	AREA NAME	Rpt Dist No	Part 1-2	Crm Cd	Crm Cd Desc	Mocodes	...	Weapon Desc	Status	Status Desc	Crm Cd 2	Crm Cd 3	Crm Cd 4	LOCATION	LAT	LON	LOCATION_CD
2010-02-20 13:50:00	13	Newton	1385	2	900	VIOLATION OF COURT ORDER	0913 1814 2000	...	unknown	AA	Adult Arrest	-1	-1	-1	300 E GAGE AV	33.9825	-118.2695	0
2010-01-14 20:45:00	13	Newton	1385	2	946	OTHER MISCELLANEOUS CRIME	0421 0906	...	unknown	AO	Adult Other	-1	-1	-1	300 E GAGE AV	33.9825	-118.2695	0
2010-01-14 20:45:00	13	Newton	1385	2	946	OTHER MISCELLANEOUS CRIME	0421	...	VERBAL THREAT	IC	Invest Cont	-1	-1	-1	300 E GAGE AV	33.9825	-118.2695	0
2010-01-15 16:15:00	13	Newton	1385	2	946	OTHER MISCELLANEOUS CRIME	0334 0432 0913	...	unknown	AO	Adult Other	-1	-1	-1	300 E GAGE AV	33.9825	-118.2695	0
2010-						THEFT FROM MOTOR VEHICLE	0244				Invest				300 E GAGE			

➤ Date Reported dimension

Τώρα, δημιουργείται η διάσταση Date Reported. Η συγκεκριμένη διάσταση περιλαμβάνει την στήλη Date Rptd. Επίσης, για την δημιουργία της διάστασης καθαρίζουμε τα επαναλαμβανόμενες τιμές Date Rptd του πίνακα `la_crimes`.

Παράλληλα, το index του νέου dataframe `date_rptd_dimension` που δημιουργήσαμε, αποθηκεύεται σε νέα στήλη προκειμένου να χρησιμοποιηθεί σαν ξένο κλειδί στην συνέχεια.

Date reported dimension

- Next up we create the dimension of the date that the crime was reported
- Particularly we take the column `Date Rptd` and clear the duplicate values
- The results are saved in a new dataframe called `date_rptd_dimension`
- We also save the index of the new dataframe as `DATE_RPTD_CD`

```
In [65]: date_rptd_dimension = la_crimes['Date Rptd'].drop_duplicates()
date_rptd_dimension = pd.DataFrame({'Date Rptd':date_rptd_dimension})
date_rptd_dimension = date_rptd_dimension.reset_index(drop = True)
date_rptd_dimension['DATE_RPTD_CD'] = date_rptd_dimension.index
date_rptd_dimension
```

```
Out[65]:
```

	Date Rptd	DATE_RPTD_CD
0	2010-02-20	0
1	2010-01-14	1
2	2010-01-15	2
3	2010-02-16	3
4	2010-04-03	4
...
3825	2020-03-21	3825
3826	2020-06-01	3826
3827	2020-06-07	3827
3828	2020-06-21	3828
3829	2020-06-28	3829

Τώρα απομένει να δημιουργήσουμε μια νέα στήλη ('DATE_RPTD_CD') στο fact table που να περιέχει τις τιμές του ξένου κλειδιού του dimension table, προκειμένου να συνδέσουμε τις διάφορες τιμές του dimension table με τα διάφορα περιστατικά. Αυτό

γίνεται εδώ και ύστερα μπορούμε να διαγράψουμε την στήλη Date Rptd από το fact table.

- We will use the index column of the new dataframe `date_rptd_dimension` as a foreign key and thus we need to add a new column to `la_crimes` containing for each incident the value of this foreign key
- After we do that, the column `Date Rptd` is no longer needed in the dataframe `la_crimes` as we have the mapping from the column `DATE_RPTD_CD` to the dimension table `date_rptd_dimension`

```
In [66]: la_crimes = pd.merge(la_crimes,
                           date_rptd_dimension,
                           on='Date Rptd',
                           how='inner')
la_crimes
```

AREA NAME	Rpt Dist No	Part 1-2	Crm Cd	Crm Cd Desc	Mocodes	...	Status	Status Desc	Crm Cd 2	Crm Cd 3	Crm Cd 4	LOCATION	LAT	LON	LOCATION_CD	DATE_RPTD_CD
Newton	1385	2	900	VIOLATION OF COURT ORDER	0913 1814 2000	...	AA	Adult Arrest	-1	-1	-1	300 E GAGE AV	33.9825	-118.2695	0	0
Newton	1385	2	930	CRIMINAL THREATS - NO WEAPON DISPLAYED	0371 0421 0906	...	AA	Adult Arrest	-1	-1	-1	300 E GAGE AV	33.9825	-118.2695	0	0
Newton	1385	2	900	VIOLATION OF COURT ORDER	0913 1814 2000	...	IC	Invest Cont	-1	-1	-1	300 E GAGE AV	33.9825	-118.2695	0	0
Central	152	2	930	CRIMINAL THREATS - NO WEAPON DISPLAYED	0421 0903	...	AA	Adult Arrest	-1	-1	-1	600 W 5TH ST	34.0502	-118.2540	13	0
Central	152	2	930	INTIMATE PARTNER - SIMPLE	0416 1244	...	AO	Adult	-1	-1	-1	1300 S	34.0501	-118.2550	15	0

➤ Date occurred dimension

Έχτερα, δημιουργείται η διάσταση Date occurred. Η συγκεκριμένη διάσταση περιλαμβάνει την στήλη DATE OCC. Επίσης, για την δημιουργία της διάστασης καθαρίζουμε τα επαναλαμβανόμενες τιμές DATE OCC του πίνακα `la_crimes`. Παράλληλα, το index του νέου dataframe `date_occ_dimension` που δημιουργήσαμε, αποθηκεύεται σε νέα στήλη προκειμένου να χρησιμοποιηθεί σαν ξένο κλειδί στην συνέχεια.

Date occurred dimension

- Next up we create the dimension of the date that the crime happened
- Particularly we take the column `DATE OCC` and clear the duplicate values
- The results are saved in a new dataframe called `date_occ_dimension`
- We also save the index of the new dataframe as `DATE_OCC_CD`

```
In [67]: date_occ_dimension = la_crimes['DATE OCC'].drop_duplicates()
date_occ_dimension = pd.DataFrame({'DATE OCC':date_occ_dimension})
date_occ_dimension = date_occ_dimension.reset_index(drop = True)
date_occ_dimension['DATE_OCC_CD'] = date_occ_dimension.index
date_occ_dimension
```

Out[67]:

	DATE OCC	DATE_OCC_CD
0	2010-02-20 13:50:00	0
1	2010-02-20 14:35:00	1
2	2010-02-20 17:00:00	2
3	2010-02-20 11:00:00	3
4	2010-02-16 08:15:00	4
...
626060	2019-12-01 06:15:00	626060
626061	2013-12-01 12:55:00	626061
626062	2019-05-12 05:36:00	626062
626063	2015-06-07 04:20:00	626063
626064	2019-07-21 08:26:00	626064

626065 rows × 2 columns

Σε αυτό το σημείο απομένει να δημιουργήσουμε μια νέα στήλη ('DATE_OCC_CD') στο fact table που να περιέχει τις τιμές του ξένου κλειδιού του dimension table, προκειμένου να συνδέσουμε τις διάφορες τιμές του dimension table 'Date occurred' με τα διάφορα περιστατικά. Αυτό ακριβώς πραγματοποιείται εδώ και ύστερα μπορούμε να διαγράψουμε την στήλη DATE OCC από το fact table.

- We will use the index column of the new dataframe `date_occ_dimension` as a foreign key and thus we need to add a new column to `la_crimes` containing for each incident the value of this foreign key
- After we do that, the column `DATE OCC` is no longer needed in the dataframe `la_crimes` as we have the mapping from the column `DATE_OCC_CD` to the dimension table `date_occ_dimension`

```
In [68]: la_crimes = pd.merge(la_crimes,
                           date_occ_dimension,
                           on='DATE OCC',
                           how='inner')
```

REA	AREA NAME	Rpt Dist No	Part 1-2	Crm Cd	Crm Cd Desc	Mocodes	...	Status Desc	Crm Cd 2	Crm Cd 3	Crm Cd 4	LOCATION	LAT	LON	LOCATION_CD	DATE_RPTD_CD	DATE_OCC_CD
13	Newton	1385	2	900	VIOLATION OF COURT ORDER	0913 1814 2000	...	Adult Arrest	-1	-1	-1	300 E GAGE AV	33.9825	-118.2695	0	0	0
13	Newton	1385	2	900	VIOLATION OF COURT ORDER	0913 1814 2000	...	Invest Cont	-1	-1	-1	300 E GAGE AV	33.9825	-118.2695	0	0	0
9	Van Nuys	943	2	928	THREATENING PHONE CALLS/LETTERS	0421	...	Invest Cont	-1	-1	-1	14100 OXNARD ST	34.1794	-118.4400	20593	0	0
13	Newton	1385	2	930	CRIMINAL THREATS - NO WEAPON DISPLAYED	0371 0421 0906	...	Adult Arrest	-1	-1	-1	300 E GAGE AV	33.9825	-118.2695	0	0	1

➤ Suspect dimension

Δημιουργούμε τη διάσταση Suspect. Η συγκεκριμένη διάσταση περιλαμβάνει την στήλη Mocodes. Επίσης, για την δημιουργία της διάστασης καθαρίζουμε τα επαναλαμβανόμενες τιμές Mocodes του πίνακα la_crimes.

Παράλληλα, το index του νέου dataframe suspect_dimension που δημιουργήσαμε, αποθηκεύεται σε νέα στήλη 'MOCODE_CD' προκειμένου να χρησιμοποιηθεί σαν ξένο κλειδί στην συνέχεια.

Suspect dimension

- Now we create the dimension of the suspect. Mocodes are activities associated with the suspect in commission of the crime
- Particularly we take the column Mocodes and clear the duplicate values
- The results are saved in a new dataframe called suspect_dimension
- We also save the index of the new dataframe as MOCODES_CD

```
In [69]: suspect_dimension = la_crimes['Mocodes'].drop_duplicates()
suspect_dimension = pd.DataFrame({'Mocodes':suspect_dimension})
suspect_dimension = suspect_dimension.reset_index(drop = True)
suspect_dimension['MOCODE_CD'] = suspect_dimension.index
suspect_dimension
```

Out [69]:

	Mocodes	MOCODE_CD
0	0913 1814 2000	0
1	0421	1
2	0371 0421 0906	2
3	0421 0903	3
4	0344	4
...
483757	0344 1236 0913 1803 0601	483757
483758	1820 1259 1251 1815 0913 0421 0356	483758
483759	0522 1251 1822 1259	483759
483760	0416 0444 0446 1402 1814 2000	483760
483761	0500 0519 0906 0913 1202	483761

483762 rows × 2 columns

Τώρα πρέπει να δημιουργήσουμε μια νέα στήλη ('MOCODE_CD') στο fact table που να περιέχει τις τιμές του ξένου κλειδιού του dimension table, προκειμένου να συνδέσουμε τις διάφορες τιμές του dimension table 'suspect_dimension' με τα διάφορα εγκληματικά περιστατικά. Αυτό ακριβώς πραγματοποιείται εδώ και ύστερα μπορούμε να διαγράψουμε την στήλη Mocodes από το fact table.

- We will use the index column of the new dataframe suspect_dimension as a foreign key and thus we need to add a new column to la_crimes containing for each incident the value of this foreign key
- After we do that, the column Mocodes is no longer needed in the dataframe la_crimes as we have the mapping from the column MOCODE_CD to the dimension table suspect_dimension

```
In [70]: la_crimes = pd.merge(la_crimes,
                         suspect_dimension,
                         on='Mocodes',
                         how='inner')
la_crimes
```

AREA NAME	Rpt Dist No	Part 1-2	Crm Cd	Crm Cd Desc	Mocodes	...	Crm Cd 2	Crm Cd 3	Crm Cd 4	LOCATION	LAT	LON	LOCATION_CD	DATE_RPTD_CD	DATE_OCC_CD	MOCODE_CD
Newton	1385	2	900	VIOLATION OF COURT ORDER	0913 1814 2000	...	-1	-1	-1	300 E GAGE AV	33.9825	-118.2695	0	0	0	0
Newton	1385	2	900	VIOLATION OF COURT ORDER	0913 1814 2000	...	-1	-1	-1	300 E GAGE AV	33.9825	-118.2695	0	0	0	0
West LA	885	2	624	BATTERY - SIMPLE ASSAULT	0913 1814 2000	...	-1	-1	-1	2900 OVERLAND AV	34.0340	-118.4188	18018	3469	103	0
Newton	1371	1	310	BURGLARY	0913 1814 2000	...	-1	-1	-1	100 W 57TH ST	33.9906	-118.2739	31858	2	720	0
Newton	1272	2	800	VIOLATION OF	0913 1814 2000	...	-1	-1	-1	300 E 56TH	33.9816	-118.2701	31041	3160	2020	0

➤ Victim dimension

Τώρα φτιάχνουμε την διάσταση που αφορά το θύμα του εγκλήματος. Η συγκεκριμένη διάσταση περιλαμβάνει τις στήλες Vict Sex και Vict Descent. Μάλιστα, η διάσταση αποθηκεύεται ως victim_dimension.

Ταυτόχρονα, δημιουργούμε από το index το ξένο κλειδί που θα χρησιμοποιήσουμε στο fact table

Victim dimension

- Next up the dimension of the victim will be made
- Particularly we take the columns Vict Sex and Vict Descent and clear the duplicate values
- The results are saved in a new dataframe called victim_dimension

```
In [71]: victim_dimension = la_crimes[['Vict Sex','Vict Descent']].drop_duplicates()
victim_dimension = victim_dimension.reset_index(drop=True)
victim_dimension['VICTIM_CD'] = victim_dimension.index
victim_dimension
```

	Vict Sex	Vict Descent	VICTIM_CD
0	M	H	0
1	F	B	1
2	M	B	2
3	F	H	3
4	M	W	4
5	F	W	5
6	F	O	6
7	M	O	7
8	M	K	8
9	F	A	9
10	M	A	10

Σε αυτό το σημείο πρέπει να δημιουργήσουμε μια νέα στήλη ('VICTIM_CD') στο fact table που να περιέχει τις τιμές του ξένου κλειδιού του dimension table, προκειμένου να συνδέσουμε τις διάφορες τιμές του dimension table 'victim_dimension' με τα διάφορα εγκληματικά περιστατικά. Αυτό ακριβώς πραγματοποιείται εδώ και ύστερα μπορούμε να διαγράψουμε τις στήλες Vict Sex και Vict Descent από το fact table.

- We will use the index column of the new dataframe victim_dimension as a foreign key and thus we need to add a new column to la_crimes containing for each incident the value of this foreign key
- After we do that, the columns Vict Sex Vict Descent is no longer needed in the dataframe la_crimes as we have the mapping from the column VICTIM_CD to the dimension table victim_dimension

```
In [72]: la_crimes = pd.merge(la_crimes,
                           victim_dimension,
                           on=['Vict Sex','Vict Descent'],
                           how='inner')
```

Opt No	Part 1-2	Crm Cd	Crm Cd Desc	Mocodes	...	Crm Cd 3	Crm Cd 4	LOCATION	LAT	LON	LOCATION_CD	DATE_RPTD_CD	DATE_OCC_CD	MOCODE_CD	VICTIM_CD
485	2	900	VIOLATION OF COURT ORDER	0913 1814 2000	...	-1	-1	300 E GAGE AV	33.9825	-118.2695	0	0	0	0	0
52	2	900	VIOLATION OF COURT ORDER	0913 1814 2000	...	-1	-1	400 E 47TH ST	34.0007	-118.2696	32620	2600	171067	0	0
169	2	901	VIOLATION OF RESTRAINING ORDER	0913 1814 2000	...	-1	-1	1100 S VERMONT AV	34.0509	-118.2916	53352	1905	211794	0	0
69	2	901	VIOLATION OF RESTRAINING ORDER	0913 1814 2000	...	-1	-1	1100 S WESTMORELAND AV	34.0508	-118.2891	53889	2294	216664	0	0
142	2	029	THREATENING PHONE	0421	...	-1	-1	14100 OXNARD	34.1704	-118.4400	20502	0	12	1	0

➤ Crime dimension

Έπειτα η δημιουργία της διάστασης του τύπου εγκλήματος. Στη συγκεκριμένη διάσταση παίρνουμε τις στήλες Crm Cd, Crm Cd Desc, Part 1-2.

Αφού ελέγξουμε ότι κάθε crime code ('Crm Cd') αντιστοιχεί σε ένα μόνο crime description και part1-2, δεν απαιτείται η δημιουργία άλλου ξένου κλειδιού.

Crime dimension

- Next up the dimension of the crime committed will be made
- Particularly we take the columns Crm Cd and Crm Cd Desc Part 1-2 and clear the duplicate values
- After checking that each crime code corresponds to exactly one crime description and either part 1 or 2 crime type, there is no extra work to do.
- The results are saved in a new dataframe called `crime_dimension`

```
In [73]: la_crimes[['Crm Cd','Crm Cd Desc','Part 1-2']].drop_duplicates()['Crm Cd Desc'].value_counts() #check
Out[73]:
RECKLESS DRIVING          1
CONTEMPT OF COURT          1
SHOTS FIRED AT INHABITED DWELLING 1
DRUNK ROLL - ATTEMPT      1
EMBEZZLEMENT, PETTY THEFT ($950 & UNDER) 1
..
BLOCKING DOOR INDUCTION CENTER 1
CREDIT CARDS, FRAUD USE ($950 & UNDER) 1
THEFT OF IDENTITY          1
VEHICLE - STOLEN            1
PURSE SNATCHING             1
Name: Crm Cd Desc, Length: 142, dtype: int64

In [74]: la_crimes[['Crm Cd','Crm Cd Desc','Part 1-2']].drop_duplicates()['Crm Cd'].value_counts() #check
Out[74]:
510    1
352    1
905    1
432    1
870    1
..
951    1
440    1
441    1
442    1
434    1
Name: Crm Cd, Length: 142, dtype: int64

In [75]: crime_dimension = la_crimes[['Crm Cd','Crm Cd Desc','Part 1-2']].drop_duplicates()
```

Επιπρόσθετα, προσθέτουμε μια επιπλέον γραμμή στο συγκεκριμένο dimension που θα αντιστοιχεί σε κανένα έγκλημα με τιμή -1 για Crm Cd. Αυτό το κάνουμε αφού τις περισσότερες φορές οι στήλες Crm Cd 2, Crm Cd 3 και Crm Cd 4 δεν περιέχουν κάποιο επιπλέον έγκλημα. Μάλιστα, σε αυτή την περίπτωση έχουμε ορίσει ότι οι στήλες αυτές παίρνουν την τιμή -1 (όχι null). Έτσι, με αυτή την προσθήκη γίνεται η αντιστοίχιση και για αυτές τις περιπτώσεις.

- Moreover, we add the crime code Crm Cd -1 as a code for no crime . This will be used for the the columns Crm Cd 2, Crm Cd 3 and Crm Cd 4 (instead of them being null) because some crime incidents only include one primary crime

```
In [76]: crime_dimension = crime_dimension.append({'Crm Cd':-1, 'Crm Cd Desc': 'NO CRIME', 'Part 1-2': np.nan}, ignore_index=True)
```

Σε αυτό το σημείο πρέπει να προσθέσουμε τους κωδικούς από τις στήλες 'Crm Cd 2', 'Crm Cd 3' και 'Crm Cd 4' στο dimension table 'crime_dimension'. Συγκεκριμένα, κάποια crime codes που αφορούν δευτερεύουσες παραβάσεις δεν υπάρχουν μέσα στο crime dimension, το οποίο δημιουργήθηκε με βάση το 'Crm Cd'

Για αυτό το λόγο δημιουργούμε ένα νέο Dataframe με τους κωδικούς εγκλημάτων που συσχετίζονται με δευτερεύουσες παραβάσεις.

- Now we have to make sure that even the Crm Cd 2 Crm Cd 3 and Crm Cd 4 codes are included in the crime dimension.
- For that we create a new dataframethat includes all the unique Crm Cd 2 Crm Cd 3 Crm Cd 4 codes

```
In [77]: crm2 = la_crimes['Crm Cd 2'].drop_duplicates()
crm3 = la_crimes['Crm Cd 3'].drop_duplicates()
crm4 = la_crimes['Crm Cd 4'].drop_duplicates()
crm2 = crm2.append(crm3, ignore_index=True).drop_duplicates()
crm2 = crm2.append(crm4, ignore_index=True).drop_duplicates()
crm2

Out[77]: 0      -1
1     998
2     930
3     900
4     940
...
142    438
143    652
144     99
145    978
146    884
Length: 147, dtype: int64

In [78]: crm_sec_codes= pd.DataFrame()
crm_sec_codes['Crm Cd'] = crm2.copy()

Out[78]: Crm Cd
0      -1
1     998
2     930
3     900
4     940
...
142    438
143    652
144     99
145    978
146    884
147 rows × 1 columns
```

Έτσι, σε αυτό το σημείο προστίθενται οι κωδικοί δευτερευουσών παραβάσεων στο dimension table 'crime_dimension'. Τέλος, για τις παραβάσεις που δεν υπάρχει κάποια περιγραφή, τις ονομάζουμε 'No information'.

- Now we can merge the crimecode dimension with the additional codes
- For the codes that we do not have any crime description, we change their Crm Cd Desc to 'No information'

```
In [79]: crime_dimension = pd.merge(crime_dimension, crm_sec_codes, on='Crm Cd', how='outer')
crime_dimension['Crm Cd Desc'] = crime_dimension['Crm Cd Desc'].fillna('No information')
crime_dimension

Out[79]: Crm Cd          Crm Cd Desc  Part 1-2
0     900  VIOLATION OF COURT ORDER    2.0
1     901  VIOLATION OF RESTRAINING ORDER    2.0
2     928  THREATENING PHONE CALLS/LETTERS    2.0
3     930  CRIMINAL THREATS - NO WEAPON DISPLAYED    2.0
4     956  LETTERS, LEWD - TELEPHONE CALLS, LEWD    2.0
...
161    972           No information    NaN
162    953           No information    NaN
163    945           No information    NaN
164     99           No information    NaN
165    978           No information    NaN
166 rows × 3 columns
```

➤ Premises dimension

Τώρα δημιουργούμε την διάσταση premises ή αλλιώς εγκαταστάσεις όπου συνέβη το έγκλημα. Για την συγκεκριμένη διάσταση χρησιμοποιούμε τις στήλες Premis Cd και Premis Desc. Επίσης, η στήλη Premis Cd χρησιμοποιείται σαν ξένο κλειδί.

Παρατηρούμε ωστόσο ότι για ένα συγκεκριμένο premis description αντιστοιχούν 3 διαφορετικά premis codes. Αφού βλέπουμε ότι το συγκεκριμένο description ονομάζεται 'RETIRED (DUPLICATE) DO NOT USE THIS CODE' αλλάζουμε το όνομα του σε 'UNKNOWN' και τους premis codes που του αντιστοιχούν σε 0. Έτσι, κάθε premis description αντιστοιχεί σε ένα premis code.

Premises dimension

- Now the dimension of the premises of the crime will be made. In detail, those are the type of structure, vehicle, or location where the crime took place.
- Particularly we take the columns Premis Cd and Premis Desc and clear the duplicate values

```
In [77]: la_crimes[['Premis Cd','Premis Desc']].drop_duplicates()['Premis Desc'].value_counts()
Out[77]: RETIRED (DUPLICATE) DO NOT USE THIS CODE    3
MTA - RED LINE - HOLLYWOOD/HIGHLAND      1
MTA - GOLD LINE - MARIACHI PLAZA        1
ENERGY PLANT/FACILITY                  1
BLUE LINE (ABOVE GROUND SURFACE TRAIN)   1
..
OFFICE BUILDING/OFFICE                 1
CONSTRUCTION SITE                      1
MTA - ORANGE LINE - WOODLEY            1
MTA - RED LINE - VERMONT/BEVERLY       1
MTA - SILVER LINE - SAN PEDRO STREET STOPS  1
Name: Premis Desc, Length: 320, dtype: int64
```

- We see that different premises codes correspond to the premis desc RETIRED (DUPLICATE) DO NOT USE THIS CODE
- Because the premise description says not to be used, we will change all those premises codes and descriptions to 0 and 'unknown' respectively

```
In [78]: indexes_premis_retired = la_crimes.loc[la_crimes['Premis Desc']=='RETIRED (DUPLICATE) DO NOT USE THIS CODE'].index
la_crimes.loc[indexes_premis_retired,'Premis Desc'] = 'unknown'
la_crimes.loc[indexes_premis_retired,'Premis Cd'] = 0
```

- Each premises code corresponds to exactly one premises description
- The results are saved in a new dataframe called premises_dimension

```
In [79]: premises_dimension = la_crimes[['Premis Cd','Premis Desc']].drop_duplicates()
```

➤ Weapon dimension

Τώρα φτιάχνεται η διάσταση weapon. Για την συγκεκριμένη διάσταση χρησιμοποιούμε τις στήλες Weapon Used Cd και Weapon Desc. Εξάλλου, η στήλη Weapon Used Cd χρησιμοποιείται σαν ξένο κλειδί που θα συνδέει dimension table με fact table.

Weapon dimension

Next up the dimension of the weapon used will be created

- Particularly we take the columns Weapon Used Cd and Weapon Desc and clear the duplicate values
- The results are saved in a new dataframe called weapon_dimension

```
In [80]: la_crimes[['Weapon Used Cd','Weapon Desc']].drop_duplicates()['Weapon Desc'].value_counts()
Out[80]: unknown                           2
ANTIQUE FIREARM                         1
KNIFE WITH BLADE OVER 6 INCHES IN LENGTH 1
REVOLVER                                1
BELT FLAILING INSTRUMENT/CHAIN          1
..
CONCRETE BLOCK/BRICK                     1
TIRE IRON                               1
BOMB THREAT                             1
AUTOMATIC WEAPON/SUB-MACHINE GUN        1
UNKNOWN TYPE CUTTING INSTRUMENT         1
Name: Weapon Desc, Length: 80, dtype: int64
```

Βλέπουμε ωστόσο, ότι διαφορετικά weapon codes αντιστοιχούν σε διαφορετικά weapon description. Αυτό συμβαίνει για το weapon description 'unknown'. Για αυτό το λόγο φτιάχνουμε όλα τα weapon codes να είναι 0, για τα weapon description που είναι 'unknown'. Με αυτό τον τρόπο κάθε weapon code αντιστοιχεί σε ένα weapon description και αντίστροφα

- We see that different weapon codes correspond to the weapon desc `unknown`
- Because we have used the weapon code 0 for null values to describe incidents where we do not know the weapon, we will change the other weapon code (222) to 0 as well.

```
In [81]: la_crimes.loc[la_crimes['Weapon Desc']=='unknown']['Weapon Used Cd'].value_counts()
```

```
Out[81]: 0    1404460
         222      1
Name: Weapon Used Cd, dtype: int64
```

```
In [82]: index_weaponcd222 = la_crimes.loc[la_crimes['Weapon Used Cd']== 222].index
la_crimes.loc[index_weaponcd222,'Weapon Desc'] = 0
```

- Each weapon code corresponds to exactly one weapon description
- The results are saved in a new dataframe called `weapon_dimension`

```
In [83]: weapon_dimension = la_crimes[['Weapon Used Cd','Weapon Desc']].drop_duplicates()
```

➤ Status dimension

Ακολουθεί η διάσταση Status. Για την συγκεκριμένη διάσταση χρησιμοποιούμε τις στήλες Status και Status Desc. Εξάλλου, η στήλη Status χρησιμοποιείται σαν ξένο κλειδί που θα συνδέει dimension table με fact table.

Αφού κάθε Status αντιστοιχεί σε ένα Status Desc, δεν απαιτείται περεταίρω δουλειά

Status dimension

- Next up the dimension of the status of the crime is made
- Particularly we take the columns `Status` and `Status Desc` and clear the duplicate values
- After checking that each status code corresponds to exactly one status description, there is no extra work to do.
- The results are saved in a new dataframe called `status_dimension`

```
In [84]: la_crimes[['Status','Status Desc']].drop_duplicates()
```

```
Out[84]:
```

	Status	Status Desc
0	AA	Adult Arrest
1	AO	Adult Other
4	IC	Invest Cont
172	JO	Juv Other
557	JA	Juv Arrest
158469	CC	UNK

```
In [85]: status_dimension = la_crimes[['Status','Status Desc']].drop_duplicates()
```

➤ Coordinate dimension

Έπειτα, δημιουργείται η διάσταση Coordinate. Η συγκεκριμένη διάσταση περιλαμβάνει τις στήλες LON και LAT. Επίσης, για την δημιουργία της διάστασης καθαρίζουμε τα επαναλαμβανόμενες τιμές (LON,LAT) του πίνακα `la_crimes`.

Παράλληλα, το index του νέου dataframe coordinates_dimension που δημιουργήσαμε, αποθηκεύεται σε νέα στήλη 'COORD_CD' προκειμένου να χρησιμοποιηθεί σαν ξένο κλειδί στην συνέχεια.

Coordinate dimension

- Next up is the dimension of the geographic coordinates of the incident
- Particularly we take the columns `LON` and `LAT` and clear the duplicate values
- The results are saved in a new dataframe called `coordinates_dimension`
- We save the index of the new dataframe as a new column `COORD_CD`

```
In [86]: coordinates_dimension = la_crimes[['LON','LAT']].drop_duplicates()
coordinates_dimension = coordinates_dimension.reset_index(drop=True)
coordinates_dimension['COORD_CD'] = coordinates_dimension.index
coordinates_dimension
```

	LON	LAT	COORD_CD
0	-118.2695	33.9825	0
1	-118.2696	34.0007	1
2	-118.2916	34.0509	2
3	-118.2891	34.0508	3
4	-118.4400	34.1794	4
...
64482	-118.6071	34.2636	64482
64483	-118.6138	34.1848	64483
64484	-118.2894	33.9088	64484
64485	-118.4209	34.2301	64485
64486	-118.4210	34.0565	64486

64487 rows × 3 columns

Σε αυτό το σημείο απομένει να δημιουργήσουμε μια νέα στήλη ('COORD_CD') στο fact table που να περιέχει τις τιμές του ξένου κλειδιού του dimension table, προκειμένου να συνδέσουμε τις διάφορες τιμές του dimension table 'coordinate_dimension' με τα διάφορα περιστατικά. Αυτό ακριβώς πραγματοποιείται εδώ και ύστερα μπορούμε να διαγράψουμε τις στήλες LON και LAT από το fact table.

- We will use the index column of the new dataframe `coordinates_dimension` as a foreign key and thus we need to add a new column to `la_crimes` containing for each incident the value of this foreign key
- After we do that, the columns `LAT` and `LON` are no longer needed in the dataframe `la_crimes` as we have the mapping from the column `COORD_CD` to the dimension table `coordinates_dimension`

```
In [87]: la_crimes = pd.merge(la_crimes,
                         coordinates_dimension,
                         on=['LON','LAT'],
                         how='inner')
la_crimes
```

Part o	Part 1-2	Crm Cd	Crm Cd Desc	Mocodes	...	Crm Cd 4	LOCATION	LAT	LON	LOCATION_CD	DATE_RPTD_CD	DATE_OCC_CD	MOCODE_CD	VICTIM_CD	COORD_CD
5	2	900	VIOLATION OF COURT ORDER	0913 1814 2000	...	-1	300 E GAGE AV	33.9825	-118.2695	0	0	0	0	0	0
5	1	440	THEFT PLAIN - PETTY (\$950 & UNDER)	0344	...	-1	300 E GAGE AV	33.9825	-118.2695	0	15	4221	4	0	0
5	1	331	THEFT FROM MOTOR VEHICLE - GRAND (\$400 AND OVER)	0344	...	-1	300 E GAGE AV	33.9825	-118.2695	68762	426	11219	4	0	0
5	1	440	THEFT PLAIN - PETTY (\$950 & UNDER)	0344	...	-1	6300 S SAN PEDRO ST	33.9825	-118.2695	33025	963	212429	4	0	0

Removing and Renaming Columns

Τώρα μπορούμε να διαγράψουμε όλες τις επιπλέον στήλες που υπάρχουν ήδη στα dimension tables αφού έχουμε δημιουργήσει τις συνδέσεις μεταξύ fact table και dimension tables μέσω των ξένων κλειδιών. Έτσι μπορούμε να δούμε την μορφή του fact table μας. Συγκεκριμένα, έχει 2114699 γραμμές και 15 στήλες.

Removing dimensions from the fact table-dataframe

- Now we can drop the extra columns that there are on our fact table
- Particularly, many columns are dropped because we already made the dimensions dataframes which contain the codes and the description for each dimension
- We are going to keep the codes of the dimensions and whenever we need some extra information, we can look up the dimension tables

```
In [88]: la_crimes = la_crimes.drop(['AREA','AREA NAME','LOCATION','Rpt Dist No','Date Rptd', 'Crm Cd Desc', 'Part 1-2',  
                                'Premis Desc', 'DATE OCC', 'Mocodes','Vict Sex', 'Vict Descent',  
                                'Weapon Desc', 'Status Desc','LON','LAT'], axis=1)
```

Out[88]:

	DR_NO	Crm Cd	Vict Age	Premis Cd	Weapon Used Cd	Status	Crm Cd 2	Crm Cd 3	Crm Cd 4	LOCATION_CD	DATE_RPTD_CD	DATE_OCC_CD	MOCODE_CD	VICTIM_CD	COOF
0	1307355	900	48	501	0	AA	-1	-1	-1	0	0	0	0	0	0
1	121321665	440	43	119	0	IC	-1	-1	-1	0	15	4221	4	0	0
2	171318398	331	34	104	0	IC	-1	-1	-1	68762	426	11219	4	0	0
3	151312312	440	33	101	0	IC	-1	-1	-1	33025	963	212429	4	0	0
4	121314885	624	16	102	400	IC	-1	-1	-1	33504	3271	42901	66	0	0
...
2114694	131704213	310	43	501	0	IC	-1	-1	-1	93185	748	185385	5461	28	
2114695	172109827	662	60	101	0	IC	-1	-1	-1	113827	1170	120122	46817	28	
2114696	161807341	940	22	501	0	IC	-1	-1	-1	105196	1060	118406	169640	28	
2114697	191917901	230	37	102	311	IC	-1	-1	-1	51106	1390	20547	45099	31	
2114698	110818769	310	49	501	0	IC	-1	-1	-1	74660	10	2991	59	34	

2114699 rows × 15 columns



Επιπρόσθετα, μετονομάζουμε τις στήλες προκειμένου να μην περιέχουν κενά και μικρούς χαρακτήρες για να διευκολύνουν την χρήση τους στη συνέχεια. Τέλος, αποθηκεύουμε τα dimension tables και το fact table σε csv αρχεία προκειμένου να τα φορτώσουμε και αποθηκεύσουμε στην βάση αργότερα.

Renaming the columns name for easier use

```
In [89]: la_crimes.rename(columns={'Crm Cd':'CRM_CD',
                               'Vict Age':'VICT_AGE',
                               'Premis Cd':'PREMIS_CD','Weapon Used Cd':'WEAPON_CD',
                               'Status':'STATUS','Crm Cd 2':'CRM_CD2',
                               'Crm Cd 3':'CRM_CD3', 'Crm Cd 4':'CRM_CD4'},
                               inplace = True)

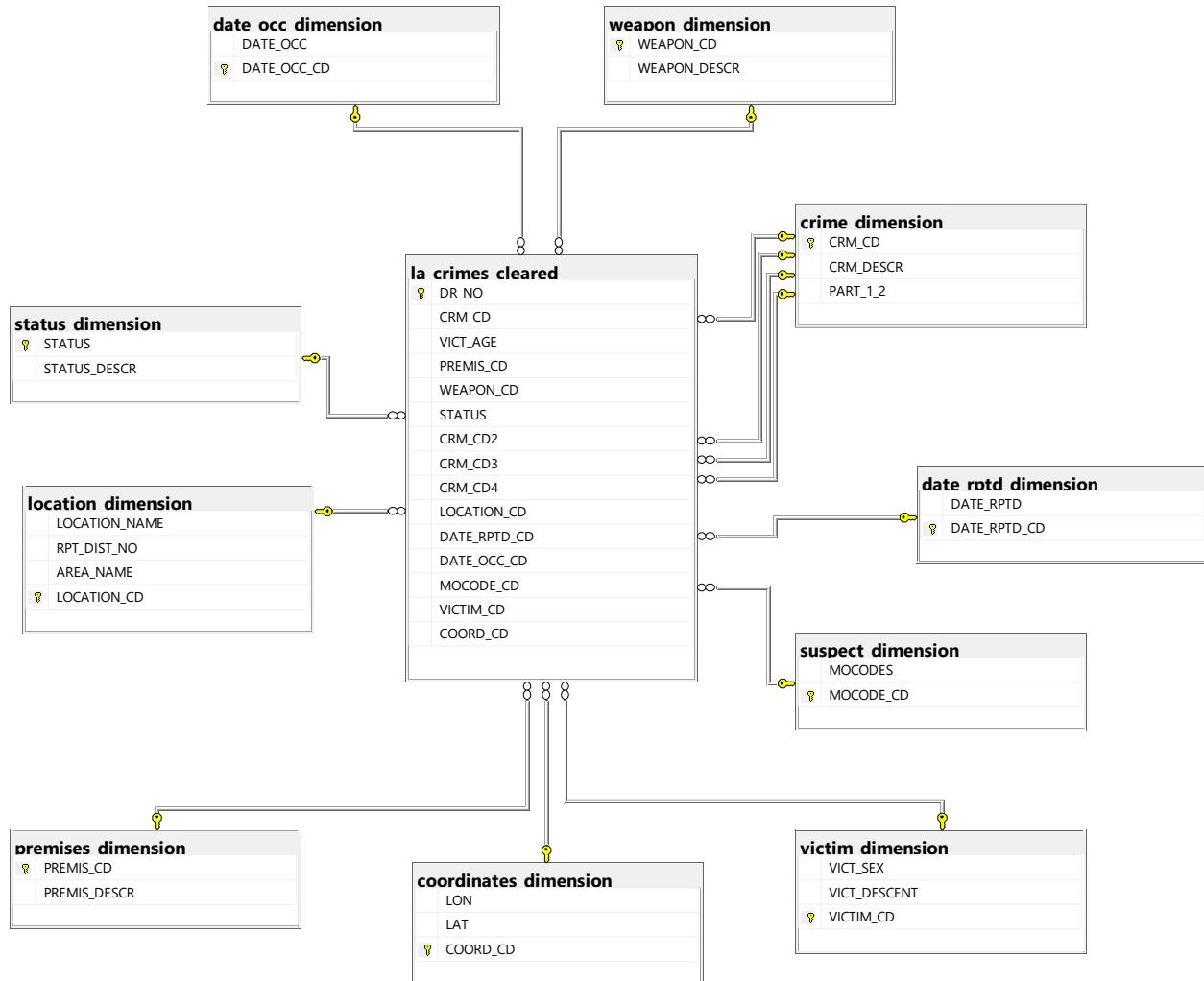
In [90]: location_dimension.rename(columns={'Rpt Dist No': 'RPT_DIST_NO','AREA NAME':'AREA_NAME','LOCATION':'LOCATION_NAME'}
                                 ,inplace = True)
crime_dimension.rename(columns={'Crm Cd': 'CRM_CD','Crm Cd Descr':'CRM_DESCR','Part 1-2':'PART_1_2'},inplace = True)
premises_dimension.rename(columns={'Premis Cd': 'PREMIS_CD','Premis Descr':'PREMIS_DESCR'},inplace = True)
weapon_dimension.rename(columns={'Weapon Used Cd': 'WEAPON_CD', 'Weapon Descr':'WEAPON_DESCR'},inplace = True)
status_dimension.rename(columns={'Status': 'STATUS','Status Descr':'STATUS_DESCR'},inplace = True)
date_occ_dimension.rename(columns={'DATE OCC': 'DATE_OCC'},inplace = True)
date_rptd_dimension.rename(columns={'Date Rptd': 'DATE_RPTD'},inplace = True)
victim_dimension.rename(columns={'Vict Sex': 'VICT_SEX','Vict Descent':'VICT_DESCENT'},inplace = True)
suspect_dimension.rename(columns={'Mocodes': 'MOCODES'},inplace = True)
```

Saving the fact table(dataframe) and dimension tables(dataframes) as csv documents

```
In [93]: la_crimes.to_csv('la_crimes_cleared.csv',index=False)
location_dimension.to_csv('location_dimension.csv',index=False)
crime_dimension.to_csv('crime_dimension.csv',index=False)
premises_dimension.to_csv('premises_dimension.csv',index=False)
weapon_dimension.to_csv('weapon_dimension.csv',index=False)
status_dimension.to_csv('status_dimension.csv',index=False)
date_occ_dimension.to_csv('date_occ_dimension.csv',index=False)
date_rptd_dimension.to_csv('date_rptd_dimension.csv',index=False)
victim_dimension.to_csv('victim_dimension.csv',index=False)
suspect_dimension.to_csv('suspect_dimension.csv',index=False)
coordinates_dimension.to_csv('coordinates_dimension.csv',index=False)
```

STAR SCHEMA

Εισάγουμε αρχικά τα δεδομένα στην αποθήκη δεδομένων με την χρήση του Microsoft SQL server management studio. Ύστερα δημιουργούμε τα διάφορα relationships δηλαδή συνδέσεις μεταξύ fact table και dimension tables μέσω των ξένων κλειδιών του fact table και των πρωτεύοντων κλειδιών των dimension table. Έτσι δημιουργείτε η αποθήκη δεδομένων μας που έχει την ακόλουθη μορφή star schema:

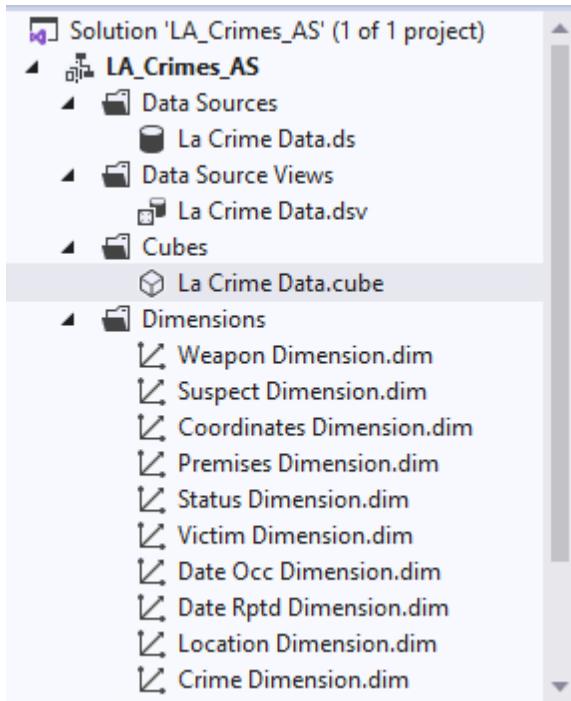


- Βλέπουμε ότι στον fact table μας η μόνη μετρική που υπάρχει είναι το VICT_AGE δηλαδή η ηλικία του θύματος (αφού συνιστά το μόνο attribute με συνεχείς αριθμητικές τιμές). Όλα τα άλλα attributes συνιστούν κλειδιά. Συγκεκριμένα, το DR_NO είναι το πρωτεύονταν κλειδί και τα υπόλοιπα είναι ξένα κλειδιά που συνδέουν τον fact table με όλα τα dimension tables.

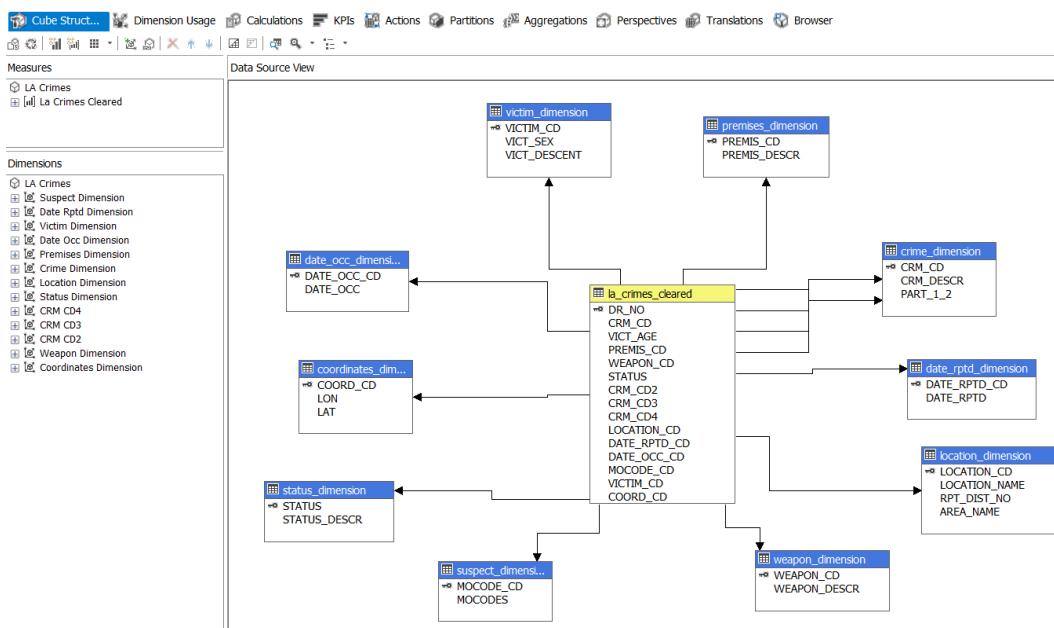
- Η διάσταση date_occ_dimension περιέχει ως πρωτεύων κλειδί τους κωδικούς DATE_OCC_CD, ο καθένας από τους οποίους αντιστοιχεί σε μια διαφορετική ημερομηνία η οποία είναι η στήλη DATE_OCC
- Η διάσταση weapon_dimension έχει ως πρωτεύων κλειδί τους κωδικούς WEAPON_CD που αντιστοιχούν σε διαφορετικό είδος όπλου (στήλη WEAPON_DESCR).
- Η διάσταση crime_dimension περιέχει ως πρωτεύων κλειδί τους κωδικούς CRM_CD, ο καθένας από τους οποίους αντιστοιχεί σε ένα διαφορετικό τύπο εγκλήματος που περιγράφεται από την στήλη CRM_DESCR. Επίσης, η στήλη PART_1_2 δείχνει την σοβαρότητα του εγκλήματος με βάση τον αμερικανικό νόμο. Το σημαντικό σε αυτήν την διάσταση είναι ότι 4 διαφορετικά ξένα κλειδιά (CRM_CD, CRM_CD2, CRM_CD3, CRM_CD4) του fact table συνδέονται μαζί της.
- Η διάσταση date_rptd_dimension περιέχει ως πρωτεύων κλειδί τους κωδικούς DATE_RPTD_CD, ο καθένας από τους οποίους αντιστοιχεί σε μια διαφορετική ημερομηνία αναφοράς η οποία είναι η στήλη DATE_RPTD
- Η διάσταση suspect_dimension έχει ως πρωτεύων κλειδί τους κωδικούς MOCODE_CD που αντιστοιχούν σε διαφορετικές δραστηριότητες του ύποπτου για το έγκλημα (στήλη MOCODES).
- Η διάσταση victim_dimension έχει ως πρωτεύων κλειδί τους κωδικούς VICTIM_CD που αντιστοιχούν σε διαφορετικό τύπο θύματος. Ο τύπος θύματος ορίζεται από το φύλο του θύματος (VICT_SEX) και την καταγωγή του (VICT_DESCENT).
- Η διάσταση coordinates_dimension περιέχει ως πρωτεύων κλειδί τους κωδικούς COORD_CD ο καθένας από τους οποίους αντιστοιχεί σε ένα μοναδικό σημείο πάνω Los Angeles. Το μοναδικό σημείο προσδιορίζεται από το γεωγραφικό πλάτος και μήκος (LAN & LON)
- Η διάσταση premises_dimension έχει ως πρωτεύων κλειδί τους κωδικούς PREMIS_CD που αντιστοιχούν σε διαφορετικό είδος εγκατάστασης (στήλη PREMIS_DESCR) όπου συνέβη το έγκλημα.
- Επίσης, η διάσταση location_dimension έχει ως πρωτεύων κλειδί τους κωδικούς LOCATION_CD και περιγράφει μοναδικές περιοχές στο Los Angeles. Οι περιοχές περιγράφονται από το LOCATION_NAME που είναι όνομα της περιοχής, το RPT_DIST_NO που είναι η υποπεριοχή εντός μιας γεωγραφικής περιοχής και το AREA_NAME που είναι η γεωγραφική περιοχή.
- Τέλος, η διάσταση status_dimension έχει ως πρωτεύων κλειδί τους κωδικούς STATUS, ο καθένας από τους οποίους αντιστοιχεί σε μια διαφορετική κατάσταση που βρίσκονται τα περιστατικά (STATUS_DESCR)

CUBE REPORTING

Ο κύβος La Crime Data.cube δημιουργήθηκε χρησιμοποιώντας την βάση δεδομένων LA Crime Data αποτελείται από measures που είναι ο αριθμός των εγκλημάτων και οι ηλικίες των θυμάτων (La Crimes Cleared Count, VICT_AGE), και από dimensions όπως παρουσιάζονται παρακάτω στο Solution Explorer του Visual Studio.



Εμφανίζεται διαγραμματικά το σχήμα και οι σχέσεις των Measures με τα διάφορα Dimensions στο Data Source View, για μια πιο ολοκληρωμένη εικόνα της δομής του κύβου.



Επίσης, μέσα στο excel μπορούμε να υπολογίσουμε όλων των ειδών συνδυασμούς όπως εδώ για παράδειγμα εδώ που βρίσκουμε ανά φύλο θύματος και ανά διαφορετικές γεωγραφικές περιοχές τα μερικά ποσοστά των καταστάσεων των περιστατικών.

1	Row Labels	La Crimes Cleared Count
2	✉ F	
3	✉ 77th Street	
4	Adult Arrest	8,81%
5	Adult Other	14,99%
6	Invest Cont	74,84%
7	Juv Arrest	1,01%
8	Juv Other	0,34%
9	UNK	0,01%
10	✉ Central	4,13%
11	✉ Devonshire	4,39%
12	✉ Foothill	3,66%
13	✉ Harbor	4,18%
14	✉ Hollenbeck	3,52%
15	✉ Hollywood	4,01%
16	✉ Mission	4,92%
17	✉ N Hollywood	5,08%
18	✉ Newton	4,64%
19	✉ Northeast	4,44%
20	✉ Olympic	4,44%
21	✉ Pacific	4,82%
22	✉ Rampart	4,23%
23	✉ Southeast	6,42%
24	✉ Southwest	7,17%
25	✉ Topanga	4,57%
26	✉ Van Nuys	4,74%
27	✉ West LA	4,21%
28	✉ West Valley	4,14%
29	✉ Wilshire	4,23%
30	✉ M	
31	✉ 77th Street	
32	Adult Arrest	8,21%
33	Adult Other	9,17%
34	Invest Cont	81,15%
35	Juv Arrest	1,24%
36	Juv Other	0,21%
37	UNK	0,01%
38	✉ Central	5,73%
39	✉ Devonshire	4,54%
40	✉ Foothill	3,83%
41	✉ Harbor	4,05%

Επιπρόσθeta, στο επόμενο παράδειγμα βλέπουμε τα συνολικά ποσοστά των περιστατικών ανά διαφορετικό status και καταγωγή του θύματος (μόνο white, black και Hispanic φαίνονται για λόγους σύγκρισης). Ανάλογες μετρικές θα φανούν καλύτερα στα παραδείγματα visualization.

1	Row Labels	La Crimes Cleared Count
2	✉ Adult Arrest	
3	B	2,09%
4	H	6,07%
5	W	2,87%
6	✉ Adult Other	
7	B	3,59%
8	H	6,58%
9	W	3,52%
10	✉ Invest Cont	
11	B	15,42%
12	H	32,83%
13	W	25,88%
14	✉ Juv Arrest	
15	B	0,17%
16	H	0,51%
17	W	0,16%
18	✉ Juv Other	0,31%
19	✉ UNK	0,00%
20	Grand Total	100,00%

Επιπλέον, εδώ βλέπουμε ένα δείγμα από τα αθροίσματα των περιστατικών ανά διαφορετικό τύπο όπλου που χρησιμοποιήθηκε και part 1 & 2 κατηγορία.

36	■ BOW AND ARROW	
37	1.0	19
38	2.0	8
39	■ BOWIE KNIFE	
40	1.0	57
41	2.0	6
42	■ BRASS KNUCKLES	
43	1.0	383
44	2.0	35
45	■ CAUSTIC CHEMICAL/POISON	
46	1.0	328
47	2.0	151
48	■ CLEAVER	
49	1.0	74
50	2.0	7
51	■ CLUB/BAT	
52	1.0	3697
53	2.0	667
54	■ CONCRETE BLOCK/BRICK	
55	1.0	498
56	2.0	389
57	■ DEMAND NOTE	
58	1.0	288
59	2.0	27
60	■ DDIRK/DAGGER	
61	1.0	108
62	2.0	16
63	■ DOG/ANIMAL (SIC ANIMAL ON)	
64	1.0	24
65	2.0	16
66	■ EXPLOXIVE DEVICE	
67	1.0	68
68	2.0	97
69	■ FIRE	
70	1.0	464
71	2.0	34
72	■ FIXED OBJECT	
73	1.0	446
74	2.0	361

Ακόμη, στην περίπτωση μας η δημιουργία επιπλέον μετρικών δεν βοηθά ιδιαίτερα αφού έχουμε στην διάθεση μας μόνο 2 μετρικές (την ηλικία του θύματος και το count που είναι για κάθε περιστατικό 1). Παρόλα αυτά, κρίθηκε σημαντική η επίδειξη δημιουργίας μιας επιπλέον μετρικής. Συγκεκριμένα, κάνουμε normalization (κανονικοποίηση) της ηλικίας έτσι ώστε όλες οι ηλικίες να παίρνουν τιμές μεταξύ του 0 και του 1. Ο τύπος της κανονικοποίησης ορίζεται ως η διαφορά κάθε συγκεκριμένης ηλικίας με την μικρότερη ηλικία προς τη διαφορά της μεγαλύτερης με την μικρότερη ηλικία:

$$\frac{X - X_{min}}{X_{max} - X_{min}}$$

Τα missing values (ηλικία 0) θα πάρουν αρνητικές τιμές τώρα. Επιπλέον, η μέγιστη ηλικία είναι 118 και η ελάχιστη 1.

The screenshot shows the Microsoft Power BI Data Model Editor interface. In the top navigation bar, the active tab is 'LA Crimes.cube [Design]'. Below the tabs, there are several icons: Cube Structure, Dimension Usage, Calculators (highlighted in blue), KPIs, Actions, Partitions, Aggregations, Perspectives, Translations, and Browser.

In the main workspace, a 'Script Organizer' pane on the left lists two items: 'CALCULATE' and 'NORMALIZED_AGE'. The 'NORMALIZED_AGE' item is selected. The right pane displays the properties for this calculated measure:

- Name:** NORMALIZED_AGE
- Parent Properties:**
 - Parent hierarchy: Measures
 - Parent member: (dropdown menu)
- Expression:** $([\text{Measures}].[\text{VICT AGE}] - 1) / (118 - 1)$
- Additional Properties:**
 - Format string: (dropdown menu)
 - Visible: True
 - Non-empty behavior: (dropdown menu)
 - Associated measure group: (dropdown menu, set to Undefined)
 - Display folder: (dropdown menu)
 - Color Expressions
 - Font Expressions

On the far left, a 'Calculation Tools' sidebar includes 'Metadata', 'Functions', and 'Templates'. Below it is a 'Measure Group' dropdown set to '<All>'. A detailed tree view under 'LA Crimes' shows various dimensions and measures, including 'La Crimes Cleared Count' and 'VICT AGE' under 'Measures'.

Εδώ βλέπουμε μερικά παραδείγματα του normalized victim age για διαφορετικές χρονικές στιγμές περιστατικών εγκλημάτων.

Row Labels	La Crimes Cleared Count	NORMALIZED_AGE
2010-01-01 00:02:00.000	1	0,205128205
2010-01-01 00:07:00.000	1	0,213675214
2010-01-01 00:25:00.000	1	0,324786325
2010-01-01 00:33:00.000	1	0,213675214
2010-01-01 00:43:00.000	1	0,316239316
2010-01-01 00:55:00.000	1	0,256410256
2010-01-01 00:59:00.000	1	0,658119658
2010-01-01 01:35:00.000	1	0,196581197
2010-01-01 01:40:00.000	1	-0,008547009
2010-01-01 01:45:00.000	1	0,273504274
2010-01-01 01:53:00.000	1	-0,008547009
2010-01-01 01:55:00.000	1	0,256410256
2010-01-01 02:10:00.000	1	0,273504274
2010-01-01 02:15:00.000	1	0,282051282
2010-01-01 02:40:00.000	1	0,162393162
2010-01-01 02:45:00.000	1	0,555555556
2010-01-01 02:55:00.000	1	0,222222222
2010-01-01 03:10:00.000	1	0,247863248
2010-01-01 03:35:00.000	1	0,478632479
2010-01-01 03:40:00.000	1	0,504273504
2010-01-01 03:50:00.000	1	0,230769231
2010-01-01 04:20:00.000	1	0,188034188
2010-01-01 04:35:00.000	1	0,179487179
2010-01-01 04:40:00.000	1	-0,008547009
2010-01-01 04:45:00.000	1	0,196581197
2010-01-01 04:55:00.000	1	-0,008547009
2010-01-01 05:55:00.000	1	0,239316239
2010-01-01 06:15:00.000	1	0,196581197

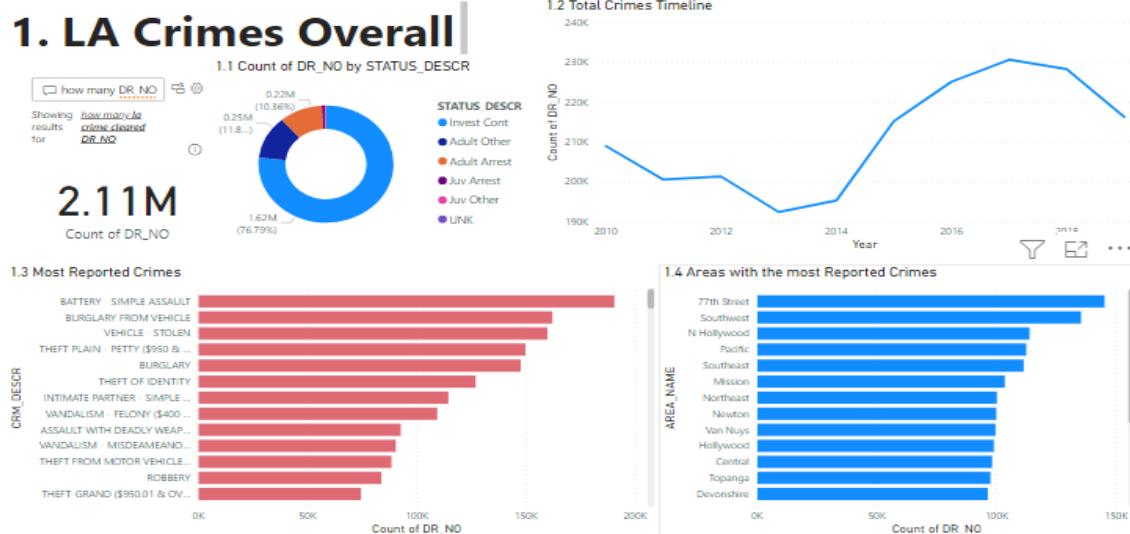
Μάλιστα, ο συγκεκριμένος κύβος χρησιμοποιήθηκε για την εξαγωγή συμπερασμάτων και αναλύσεων στο εργαλείο Power BI.

VISUALISATIONS

Σε αυτό το σημείο θα δούμε κάποιες οπτικοποιήσεις των δεδομένων μας. Αναλυτικά, έγινε χρήση του προγράμματος powerBI για τις κύριες οπτικοποιήσεις των δεδομένων μας. Επίσης, οι τελευταίες πιο εξειδικευμένες οπτικοποιήσεις έγιναν με python στο περιβάλλον jupyter notebook.

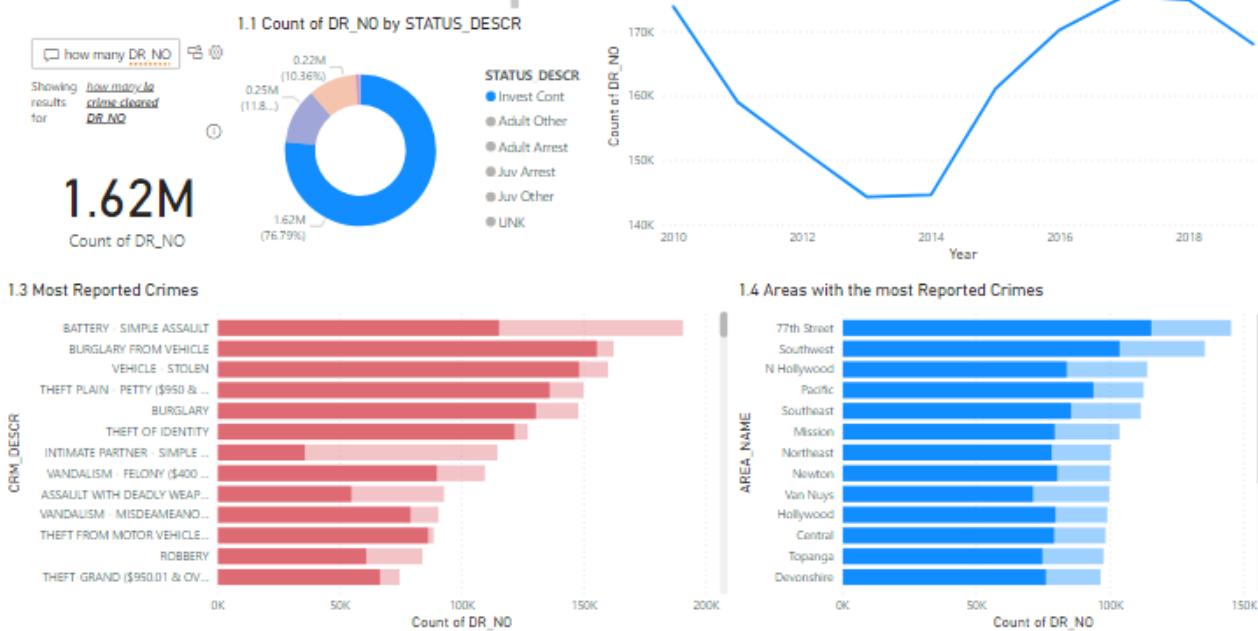
LA Crimes Overall

Συνολικά καταγράφηκαν περίπου 2,11 εκατομμύρια περιστατικά εγκλήματος στο Λος Άντζελες την περίοδο 2010 έως 2019. Στο διάγραμμα 1.3 παρουσιάζονται τα πιο συνηθισμένα εγκλήματα όπως είναι η χειροδικία/επίθεση, η διάρρηξη οχήματος, η κλοπή οχήματος ή κλοπή αντικειμένων συνολικής αξίας κάτω των 950 δολαρίων, και η διάρρηξη. Στο διάγραμμα 1.1 βλέπουμε ότι το ποσοστό των υποθέσεων για τα οποία συνεχίζεται η έρευνα και παραμένουν ανοιχτές είναι το 76,79% των εγκλημάτων, ενώ μόνο το 23,21% έχουν κλείσει. Ακόμη, στο διάγραμμα 1.2 παρατηρούμε ότι υπήρχε σταδιακή μείωση των εγκλημάτων από το 2010 έως το 2013 όπου καταγράφηκαν μόλις περίπου 190 χιλιάδες εγκλήματα. Από το 2013 μέχρι το 2017 τα εγκλήματα αυξάνονταν σταθερά και από το 2017 μέχρι το 2019 υπήρχε πάλι μια μικρή βελτίωση. Τέλος, στο διάγραμμα βλέπουμε τις περιοχές με την υψηλότερη εγκληματικότητα, όπως: η οδός 77th Street, η νοτιοδυτική περιοχή, η βόρεια πλευρά του Χόλυγουντ και η περιοχή Pacific.



Μάλιστα, μπορούμε να δούμε στα γραφήματα αποκλειστικά το ποσοστό των υποθέσεων που δεν έχουν λυθεί, στα δημοφιλέστερα εγκλήματα και την περιοχή όπου διαπράχθηκαν, καθώς και το αντίστοιχο χρονοδιάγραμμα.

1. LA Crimes Overall



Crimes analysis based on sex of victim

Στο διάγραμμα 2.1 φαίνεται ότι τα ποσοστά ανδρών και γυναικών που έπεσαν θύμα εγκλήματος είναι μοιρασμένα. Στο διάγραμμα 2.2 παρατηρούμε μάλιστα τα ποσοστά αυτά σε καθένα από τα πιο συχνά εγκλήματα του Λος Άντζελας. Στο διάγραμμα 2.3 βλέπουμε ότι τα συνηθέστερα εγκλήματα με θύμα άντρα είναι εγκλήματα όπως η επίθεση με φονικό αντικείμενο, η κλοπή ταυτότητας, ο βανδαλισμός και άλλα. Στο διάγραμμα 2.4 βλέπουμε ότι τα συνηθέστερα εγκλήματα με θύμα γυναίκα είναι εγκλήματα όπως η χειροδοκία/επίθεση, η χειροδοκία/επίθεση από ερωτικό σύντροφο, η διάρρηξη οχήματος και άλλα.

2. Crimes Based on Sex of the Victim

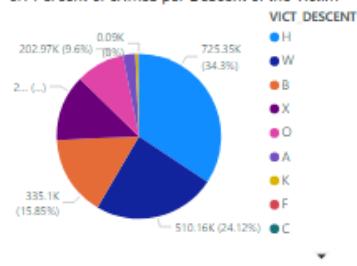


Crimes analysis based on descent of victim

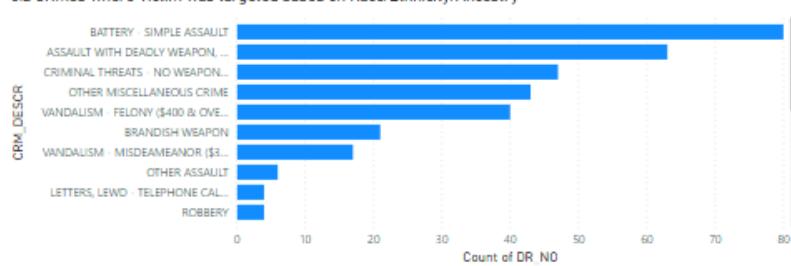
Στο διάγραμμα 3.1 παρουσιάζονται ποσοστά όπου το θύμα του εγκλήματος ήταν συγκεκριμένης καταγωγής. Οι κατηγορίες «White», «Hispanic» και «Black» είναι οι επικράτησες κατηγορίες θυμάτων με βάση την καταγωγή. Στο διάγραμμα 3.2 βλέπουμε το ποσοστό τους συχνότερους τύπους εγκλημάτων στα οποία το θύμα στοχοποιήθηκε λόγω της καταγωγής του. Τέλος, στα διαγράμματα 3.3, 3.4, 3.5 βλέπουμε τα συχνότερα εγκλήματα με θύμα καταγωγής «White», «Black» και «Hispanic» αντίστοιχα.

3. Crimes Based on Victim's Decent

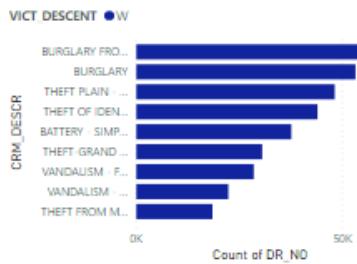
3.1 Percent of crimes per Descent of the Victim



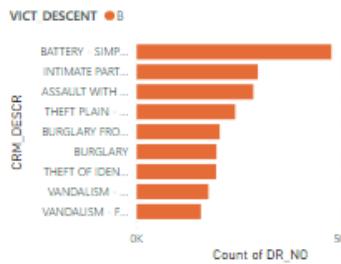
3.2 Crimes where Victim was targeted based on Race/Ethnicity/Ancestry



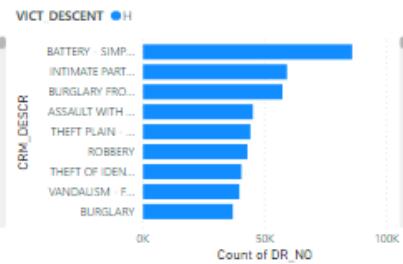
3.3 Top Crimes where the Victim was White



3.4 Top Crimes where the Victim was Black



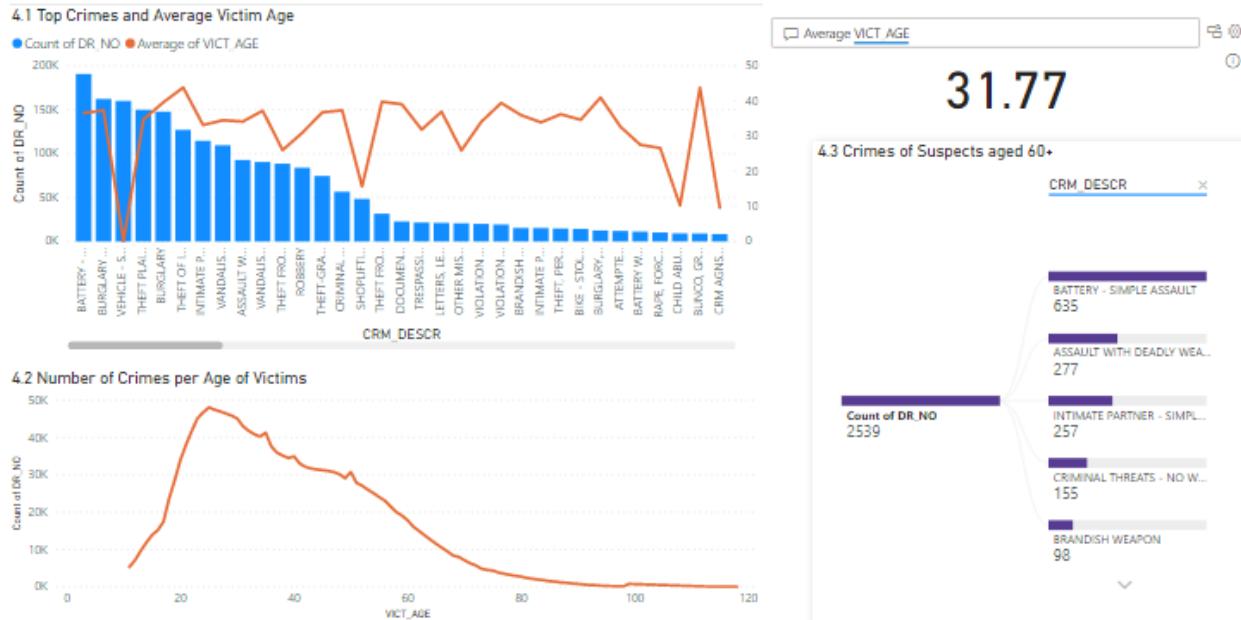
3.5 Most Crimes where the Victim was Hispanic



Crimes analysis based on age of the victim or suspect

Στο διάγραμμα 4.1 παρουσιάζονται τα συχνότερα εγκλήματα και ο μέσος ορός ηλικίας του θύματος σε καθένα από αυτά. Συνολικά, ο μέσος ορός ηλικίας των θυμάτων οποιουδήποτε εγκλήματος είναι στα 31 χρόνια. Στο διάγραμμα 4.2 εμφανίζεται ο αριθμός των θυμάτων οποιουδήποτε εγκλήματος ανά ηλικία. Παρατηρούμε ότι η ηλικία όπου παρατηρείται ο μεγαλύτερος αριθμός θυμάτων είναι στα 25 χρόνια. Ακόμη, στο διάγραμμα 4.3 παρουσιάζονται εγκλήματα που διαπράχθηκαν από ηλικιωμένους (60 χρονών και άνω).

4. Crimes based on Age of Victim or Suspect

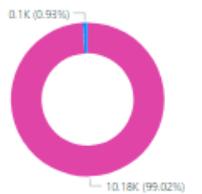


Crime of rape

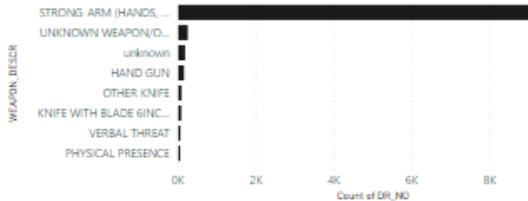
Στο διάγραμμα 5.1 γίνεται διακριτό ότι το γυναικείο φύλο είναι αυτό που στη συντριπτική πλειοψηφία έπεσε θύμα βιασμού (99.02%). Στο διάγραμμα 5.2 βλέπουμε τα συνηθέστερα όπλα που χρησιμοποιήθηκαν σε εγκλήματα τέτοιου τύπου. Δυνατά χέρια, πιστόλι χειρός, μαχαίρι είναι τα συνηθέστερα αν εξαιρέσουμε τις περιπτώσεις όπου δεν γνωρίζουμε το όπλο που χρησιμοποιήθηκε. Το 48.82% των υποθέσεων αυτών μένουν ακόμη ανοιχτές(5.3). Τα μέρη στα οποία συνέβησαν οι περισσότεροι βιασμοί είναι σε μονοκατοικία η πολυκατοικία, στον δρόμο, σε ξενοδοχείο σε όχημα (όπου συνεπιβάτης ήταν το θύμα) και άλλα (5.4). Έχουν καταγραφεί 10,28 χιλιάδες εγκλήματα βιασμού στα οποία ο μέσος ορός ηλικίας του θύματος ήταν στα 26 με 27 χρόνια.

5. Crimes of Rape

5.1 Percentage of Crimes for each Victim Sex



5.2 Most used Weapons



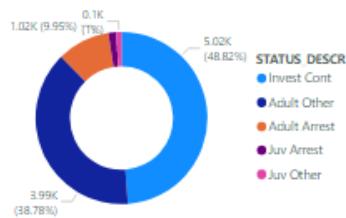
Number of Rape incidents

how many DR_NO with CRM_DESCR RAPE_FORCEABLE

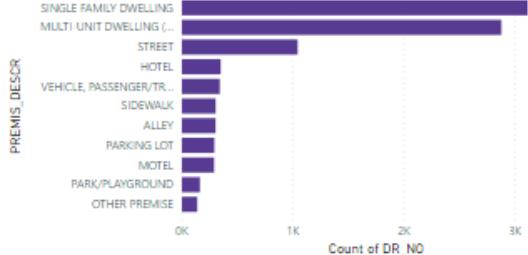
Showing how many [la crime cleared](#) results DR_NO with CRM_DESCR RAPE_FORCEABLE for

10.28K
Count of DR_NO

5.3 Percentage of Crimes per Status of Case



5.4 Most usual Premises where Crime was occurred



Average Age of Victim

average VICT_AGE for CRM_DESCR RAPE_FORCEABLE

26.57
Average of VICT_AGE

Battery – Simple Assault

Το έγκλημα της χειροδικίας/επίθεσης ήταν το συνηθέστερο έγκλημα που καταγράφηκε, με 190,54 χιλιάδες περιστατικά. Τα όπλα που χρησιμοποιήθηκαν περισσότερο παρουσιάζονται στο διάγραμμα 6.1. Στο διάγραμμα 6.2 βλέπουμε επίσης τα πιο συχνά μέρη όπου συνέβη. Κι εδώ επίσης, η πλειοψηφία των υποθέσεων παραμένουν ανοιχτές (60,44%).

6. Crimes of Battery - Simple Assault

6.1 Most usual Weapons used

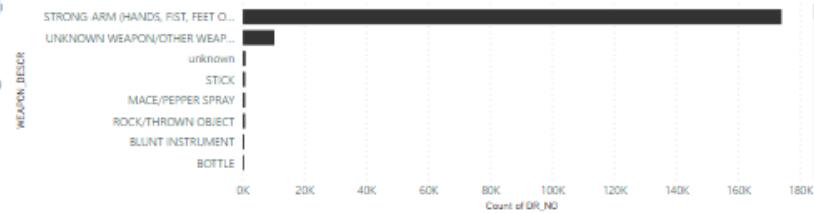
how many DR_NO CRM_DESCR BATTERY_SIMPLE ASSAULT

Showing results for how many [la crime cleared](#) DR_NO CRM_DESCR BATTERY_SIMPLE ASSAULT

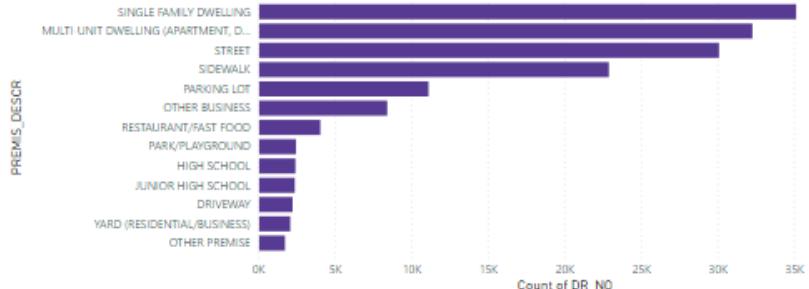
190.54K

Count of DR_NO

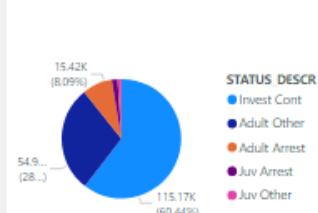
6.1 Most usual Weapons used



6.2 Most usual Premises where crime occurred



6.3 Percentage of Crimes per Status of Case

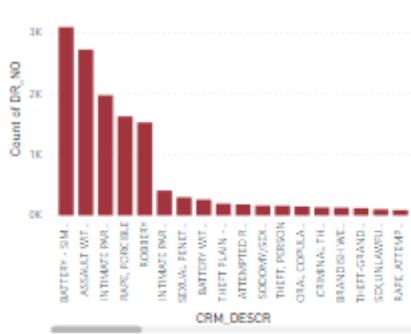


Other

Άλλα ενδιαφέροντα αποτελέσματα εξάχθηκαν από εγκλήματα που διαπράχθηκαν στα οποία ο ύποπτος ήταν υπό την επήρεια ουσιών (αλκοόλ ή ναρκωτικά) όπως βλέπουμε στο διάγραμμα 7.1, ενώ υπήρχαν μάλιστα παράξενες περιπτώσεις στις οποίες ο ύποπτος έδρασε ενώ ήταν ντυμένος με κάποιο χαρακτηριστικό κοστούμι φανταστικού χαρακτήρα (7.2)

8. Other

8.1 Suspect was intoxicated or drunk



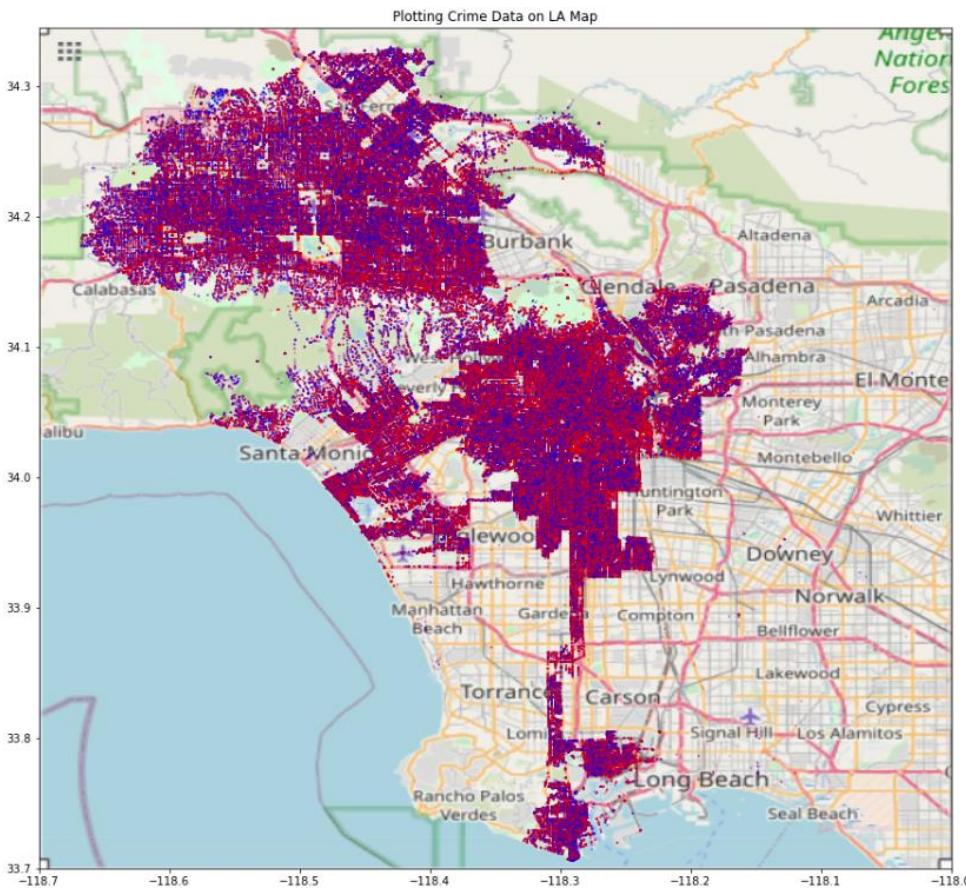
8.2 Incidents where the suspect was dressed in a costume(Spiderman, Darth Vader etc.)



Crime visualization on map

Επόμενη περίπτωση οπτικοποίησης συνιστά η αναπαράσταση των περιστατικών πάνω στον χάρτη

```
In [80]: from matplotlib import colors # importing colors
BBox = (-118.7000, -118, 33.7000, 34.3450) # coordinates of map
ruh_m = plt.imread('lamap.png') #importing the map
fig,ax1 = plt.subplots(figsize = (16,14))
cmap = colors.ListedColormap(['red','blue'],['orange','purple','c','aqua']) # create new colormap
ax1.scatter(la_crimes.LON, la_crimes.LAT, zorder=1, alpha= 0.2, cmap=cmap,
            c=la_crimes.PART1_2, s=1)
ax1.set_title('Plotting Crime Data on LA Map')
ax1.set_xlim(BBox[0],BBox[1])
ax1.set_ylim(BBox[2],BBox[3])
ax1.imshow(ruh_m, zorder=0, extent = BBox, aspect= 'equal')
```



Κάθε περιστατικό που συνέβη μέσα στην δεκαετία 2010 έως 2019 αποτελεί μία τελεία. Το κόκκινο χρώμα δηλώνει ότι το περιστατικό ήταν part 1 offense δηλαδή πιο σοβαρό έγκλημα όπως burglary, robbery, rape και manslaughter και άλλα. Το μπλε χρώμα σημαίνει ότι το περιστατικό ήταν part 2 offense δηλαδή λιγότερο σοβαρό έγκλημα, για παράδειγμα theft of identity, trespassing, vandalism, simple assault και άλλα. Τέλος, φαίνεται ξεκάθαρα ότι στο κέντρο του Los Angeles έχουν πραγματοποιηθεί πολύ περισσότερα εγκλήματα αφού οι τελείες εκεί είναι πολύ πιο πυκνές.

Delay of report per crime type

Τώρα θα εξετάσουμε την χρονική καθυστέρηση μεταξύ του χρόνου που έγινε το έγκλημα και του χρόνου που αυτό αναφέρθηκε.

- Υπολογίζουμε αρχικά την χρονική διαφορά για κάθε μεμονωμένο περιστατικά αφού αφαιρούμε την ημερομηνία που έγινε το έγκλημα από την ημερομηνία που αυτό αναφέρθηκε.

```
In [85]: la_crimes['TIME_DIFF'] = la_crimes['DATE_RPTD'].dt.date - la_crimes['DATE_OCC'].dt.date
la_crimes['TIME_DIFF']

Out[85]: 0      0 days
1      1 days
2      0 days
3      0 days
4      1 days
...
2114694 0 days
2114695 1 days
2114696 0 days
2114697 1 days
2114698 0 days
Name: TIME_DIFF, Length: 2114699, dtype: timedelta64[ns]

  • Changing the datatype of TIME_DIFF to float

In [96]: la_crimes['TIME_DIFF'] = la_crimes['TIME_DIFF'].astype('timedelta64[D]')
```

- Ύστερα, ομαδοποιούμε τα περιστατικά ανά τύπο εγκλήματος και παίρνουμε τον μέσο όρο της χρονικής καθυστέρησης ανά τύπο εγκλήματος.

- Getting the mean of time delay per crime type

```
In [111]: crimes_delay = la_crimes[['TIME_DIFF','CRM_DESC']].groupby(by='CRM_DESC').mean()
crimes_delay = crimes_delay.reset_index()
crimes_delay = crimes_delay.sort_values(by='TIME_DIFF', ascending=False)
crimes_delay
```

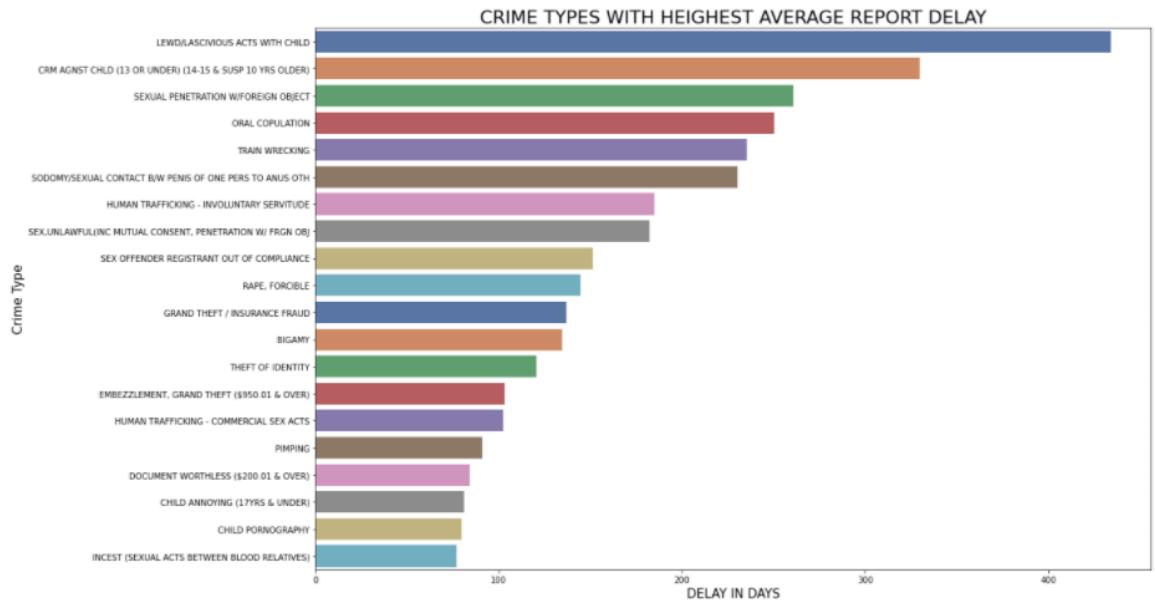
	CRM_DESC	TIME_DIFF
80	LEWD/LASCIVIOUS ACTS WITH CHILD	434.185654
40	CRM AGNST CHLD (13 OR UNDER) (14-15 & SUSP 10 ...	329.811556
104	SEXUAL PENETRATION W/FOREIGN OBJECT	260.701994
84	ORAL COPULATION	250.438720
130	TRAIN WRECKING	235.500000
...
81	LYNCHING	0.288889
11	BIKE - ATTEMPTED STOLEN	0.205128
127	TILL TAP - ATTEMPT	0.000000
65	FIREARMS TEMPORARY RESTRAINING ORDER (TEMP FIR...	0.000000
64	FIREARMS RESTRAINING ORDER (FIREARMS RO)	0.000000

142 rows × 2 columns

- Εδώ βλέπουμε τους τύπους εγκλημάτων με τον μεγαλύτερο μέσο χρόνο καθυστέρησης για αναφορά

```
In [130]: fig_dims = (18, 12)
fig, ax = plt.subplots(figsize=fig_dims)
sns.barplot(x='TIME_DIFF', y='CRM_DESC', data=crimes_delay[:20], palette='deep')
plt.title('CRIME TYPES WITH HIGHEST AVERAGE REPORT DELAY', fontdict = {'size': 22})
#ax.set_xlim(0,100)
plt.ylabel('Crime Type', fontdict = {'size': 15})
plt.xlabel('DELAY IN DAYS',fontdict = {'size': 15})
```

Out[130]: Text(0.5, 0, 'DELAY IN DAYS')

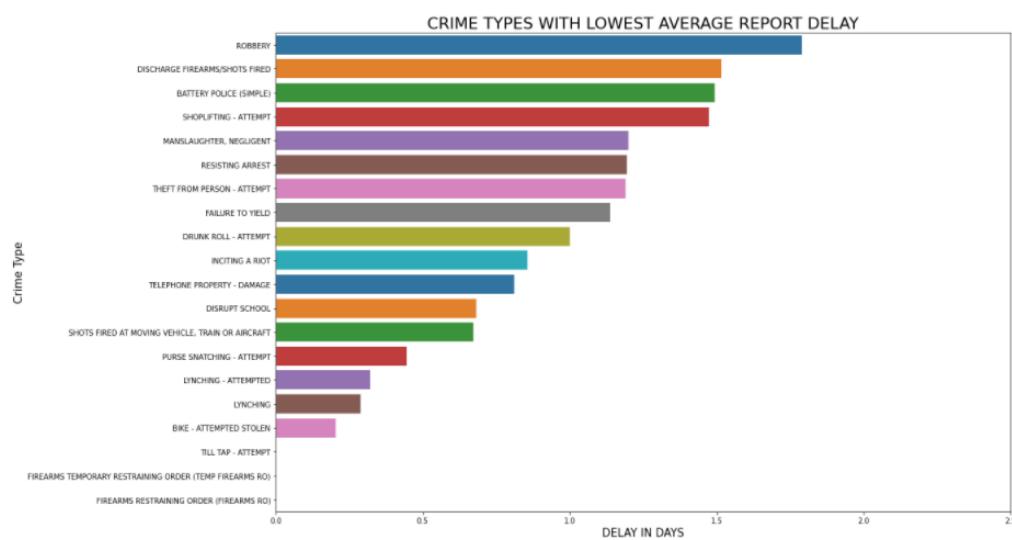


Βλέπουμε ότι τα περισσότερα εγκλήματα αφορούν είτε σεξουαλικά παραπτώματα όπως rape, sodomy, oral copulation etc. είτε το θύμα δεν γνώριζε για το παράπτωμα, όπως theft of identity, bigamy, child offenses (όπου το θύμα-παιδί δεν είχε επίγνωση ότι η συγκεκριμένη δράση είναι έγκλημα) etc. Τέλος φαίνεται ότι όλα από τα παραπάνω εγκλήματα έχουν μέσο χρόνο καθυστέρησης για αναφορά πάνω από 80 ημέρες

- Εδώ βλέπουμε τους τύπους εγκλημάτων με τον μικρότερο μέσο χρόνο καθυστέρησης για αναφορά

```
In [131]: fig_dims = (18, 12)
fig, ax = plt.subplots(figsize=fig_dims)
sns.barplot(x='TIME_DIFF', y='CRM_DESC', data=crimes_delay[-20:], palette='tab10')
plt.title('CRIME TYPES WITH LOWEST AVERAGE REPORT DELAY', fontdict={'size': 22})
ax.set_xlim(0,2.5)
plt.ylabel('Crime Type', fontdict={'size': 15})
plt.xlabel('DELAY IN DAYS', fontdict={'size': 15})

Out[131]: Text(0.5, 0, 'DELAY IN DAYS')
```



Όλα τα συγκεκριμένα είδη εγκλήματος έχουν μέσο χρόνο καθυστέρησης αναφοράς λιγότερο από 2 μέρες. Παρατηρούμε ότι τα εγκλήματα αφορούν είτε κάποιο έγκλημα κλεψιάς όπως robbery, shoplifting, bike steal etc. είτε κάποιο έγκλημα διατάραξης της κοινής ησυχίας είτε κάποιο έγκλημα έναντι αστυνομικού όπως battery police, resisting arrest.

Crimes percentages when suspect uses force

Επίσης, πραγματοποιήσαμε και μερικά διαγράμματα χρησιμοποιώντας εξειδικευμένες περιπτώσεις του suspect (μέσω της στήλης mpcodes).

- Αφού αναζητήσαμε στο έξτρα αρχείο με τα mpcodes των κωδικό που αντιστοιχεί σε force used (mocode 0400), ομαδοποιήσαμε τα περιστατικά ανά crime type και πήραμε τα ποσοστά του καθενός type σχετικά με το σύνολο των crimes με τον συγκεκριμένο mocode.

```
0399    Vehicle to Vehicle shooting
0400    Force used
0401    Bit
```

Let's find all the incidents in which force was used by the suspect. According to the Mocodes file which explains all codes, code 0400 defines the suspect using force. We see that there are 137845 incidents reported

In [204]:	crimes_force = la_crimes.loc[la_crimes['MOCODES'].str.contains('0400')]
Out [204]:	DR_NO DATE_RPTD DATE_OCC AREA_NAME DISTR_NO PART1_2 CRM_CD CRM_DESC MOCODES VICT_AGE ... VICT_DESC PREMIS
	3 90631215 2010-01-05 2010-01-05 01:50:00 Hollywood 646 2 900 VIOLATION OF COURT ORDER 1100 0400 1402 47 ... W S
	4 100100501 2010-01-03 2010-01-02 21:00:00 Central 176 1 122 RAPE, ATTEMPTED 0400 47 ... H
	8 100100510 2010-01-09 2010-01-09 02:30:00 Central 171 1 230 ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT 0400 0416 30 ... H PARKII
	10 100100521 2010-01-14 2010-01-14 14:45:00 Central 118 2 624 BATTERY - SIMPLE ASSAULT 0400 0428 2000 38 ... B S

- Let's sum up the incidents with the same crime code

In [205]:	counts = crimes_force['CRM_DESC'].value_counts()
Out [205]:	BATTERY - SIMPLE ASSAULT 48773 INTIMATE PARTNER - SIMPLE ASSAULT 28953 ROBBERY 15385 ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT 13777 INTIMATE PARTNER - AGGRAVATED ASSAULT 2938 ... TRAIN WRECKING 1 BRIBERY 1 CONTRIBUTING 1 CHILD PORNOGRAPHY 1 UNAUTHORIZED COMPUTER ACCESS 1 Name: CRM_DESC, Length: 109, dtype: int64

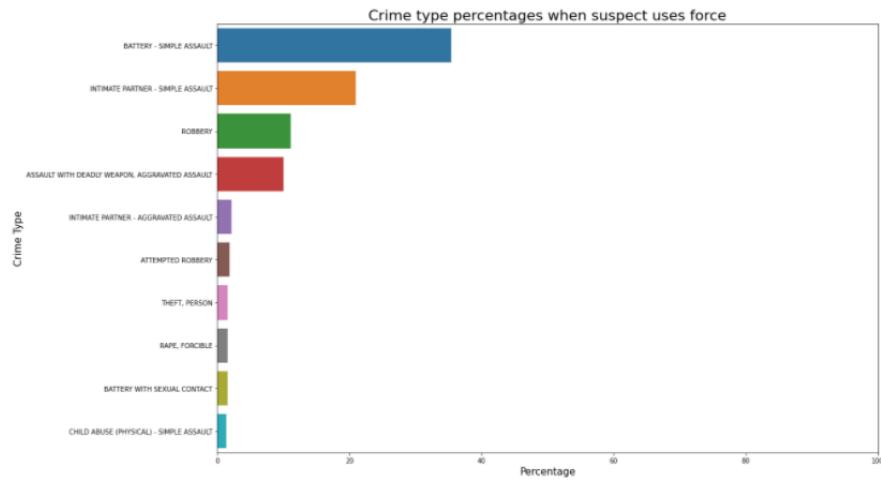
Now we will get the percentage for each crime type where the suspect uses force

```
In [206]: sum = counts.values.sum()
percentage = []
for i in counts.values:
    percentage.append( (i / sum) * 100 )
percentage
```

- Και τώρα μπορούμε να δούμε το σχήμα:

```
In [207]: fig_dims = (18, 12)
fig, ax = plt.subplots(figsize=fig_dims)
sns.barplot(x=percentage[:10], y=counts.index[:10], )
plt.title('Crime type percentages when suspect uses force', fontdict={'size': 22})
ax.set_xlim(0,100)
plt.ylabel('Crime Type', fontdict={'size': 15})
plt.xlabel('Percentage', fontdict={'size': 15})

Out[207]: Text(0.5, 0, 'Percentage')
```



Φαίνεται ότι η πλειοψηφία των εγκλημάτων σχετίζονται με κάποιο είδος assault. Κοντά στο 40% αφορά battery- simple assault αφού αυτό συνιστά και ένα από τα πιο συχνά εγκλήματα

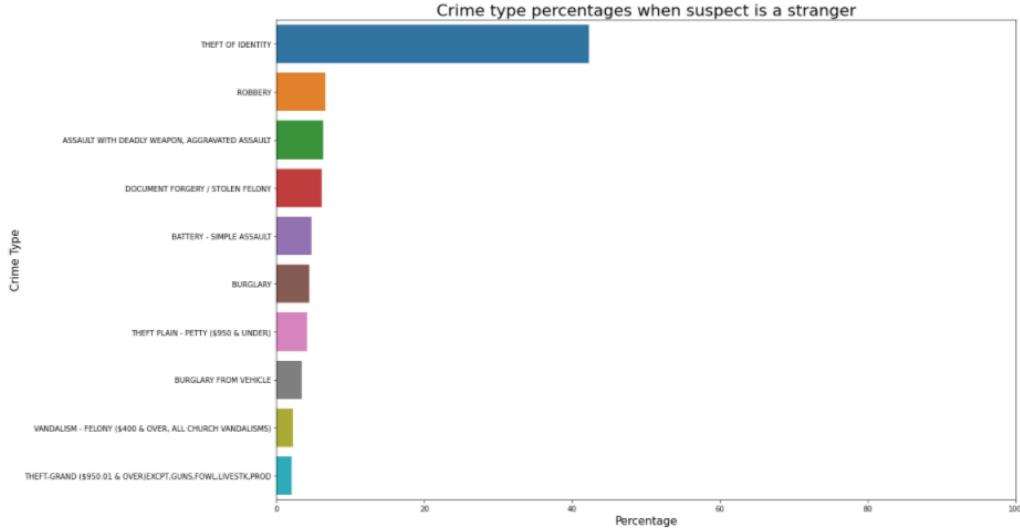
Crimes percentages when suspect is a stranger

Παρόμοια πραγματοποιήσαμε και το σχήμα με τα ποσοστά των τύπων εγκλημάτων όταν ο suspect είναι stranger

- Now we can see the plot

```
In [203]: fig_dims = (18, 12)
fig, ax = plt.subplots(figsize=fig_dims)
sns.barplot(x=percentage[:10], y=counts.index[:10], )
plt.title('Crime type percentages when suspect is a stranger', fontdict ={'size': 22})
ax.set_xlim(0,100)
plt.ylabel('Crime Type', fontdict ={'size': 15})
plt.xlabel('Percentage',fontdict ={'size': 15})

Out[203]: Text(0.5, 0, 'Percentage')
```



Παρατηρούμε ότι κοντά στο 50% των περιπτώσεων αφορά παράπτωμα theft of identity που συμβαίνει όταν κάποιος παριστάνει ότι είναι κάποιος άλλος.

Σημείωση: παρουσιάστηκαν μόνο τα είδη εγκλήματος με σημαντικά ποσοστά.

DATA MINING

Τα δεδομένα της αποθήκης θα χρησιμοποιηθούν τώρα για διάφορες λειτουργίες εξόρυξης δεδομένων. Λεπτομερώς, χρησιμοποιήθηκαν οι λειτουργίες εξόρυξης της συσταδοποίησης και της πρόβλεψης. Στη συσταδοποίηση, όπως θα δούμε στη συνέχεια, υλοποιήθηκαν 3 διαφορετικά μοντέλα και στη πρόβλεψη χρησιμοποιήθηκαν τα δέντρα αποφάσεων και οι επεκτάσεις τους για την πρόβλεψη του τύπου εγκλήματος. Τέλος, όλες τις λειτουργίες εξόρυξης δεδομένων πραγματοποιήθηκαν μέσα στο περιβάλλον του jupyter notebook, χρησιμοποιώντας βιβλιοθήκες της Python.

Clustering

Σύμφωνα με την Margaret H. Dunham (2004) (βιβλίο μαθήματος), η συσταδοποίηση είναι η διαδικασία οργάνωσης των δεδομένων σε ομάδες μη προκαθορισμένες. Μάλιστα, στοιχεία που ανήκουν στην ίδια συστάδα έχουν ομοιότητες βάσει των χαρακτηριστικών τους και στοιχεία διαφορετικών συστάδων δεν είναι όμοια.

Geographic coordinate clustering

Στο πρώτο μοντέλο που δημιουργήσαμε, θα ασχοληθούμε με την συσταδοποίηση μόνο με βάση την τοποθεσία δηλαδή τις γεωγραφικές συντεταγμένες όπου πραγματοποιήθηκαν τα διάφορα εγκλήματα. Αυτό σημαίνει ότι θα δημιουργηθούν διαφορετικά clusters που θα ομαδοποιούν εγκλήματα που πραγματοποιήθηκαν σε κοντινά σημεία. Επίσης, όπως θα δούμε, χρησιμοποιήθηκε ο αλγόριθμος k-means clustering.

- Ξεκινάμε, εισάγοντας τις απαραίτητες βιβλιοθήκες. Επίσης, παίρνουμε μόνο τις στήλες γεωγραφικού μήκους και πλάτους και αφαιροούμε τα missing values (με γεωγραφικό πλάτος και μήκος 0) για να υπάρχουν μόνο γεωγραφικά δεδομένα του Los Angeles.

CLUSTERING

- First data mining analysis will be clustering of different dimensions
- We start by importing the right libraries

```
In [64]: import folium
import json
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import plotly.express as px
from kmodes.kmodes import KModes
from yellowbrick.cluster import KElbowVisualizer
from matplotlib import colors # importing colors
```

- Next up we get the latitudes and longitudes in a separate dataframe
- We cut the incidents that have 0 for longitude/ latitude as they are missing values

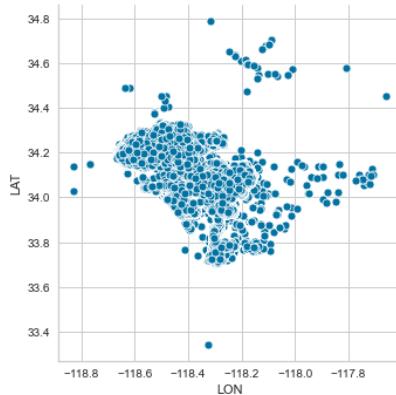
```
In [136]: location = la_crimes[['LAT','LON']].copy()
location = location.loc[(location['LAT'] != 0) & (location['LON'] != 0)]
location = location.reset_index(drop=True)
```

- Στη συνέχεια, οπτικοποιούμε τα δεδομένα για να πάρουμε μια πρώτη εικόνα. Τα δεδομένα φαίνεται να απεικονίζουν το Los Angeles

- Firstly we plot the data of longitude and latitude
 - We clearly see that the data has taken the shape of LA

```
In [66]: sns.relplot(y='LAT', x='LON', data=location)

Out[66]: <seaborn.axisgrid.FacetGrid at 0x1af53392850>
```



- Τώρα απεικονίζουμε ένα δείγμα στο χάρτη του Los Angeles για να πάρουμε μια καλύτερη εικόνα.

- Now we plot a random sample of the data onto a real Los Angeles map in order to get an idea of the crime distribution

```
In [67]: fig = px.scatter_mapbox(location.sample(100000), lat="LAT", lon="LON",
                           color_discrete_sequence=["fuchsia"], zoom=3, height=300)
fig.update_layout(mapbox_style="open-street-map")
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```



- Αλλάζουμε τα δεδομένα χρησιμοποιώντας StandardScaler() προκειμένου όλα τα δεδομένα να έχουν την διακύμανση 1 και μέσο όρο 0. Αυτό το βήμα είναι υποχρεωτικό για την χρήση του kmeans στη συνέχεια.

- We will use StandardScaler() to transform the data so that all features have equal variance.
- This is a necessary step for all clustering analysis

```
In [137]: scaler = StandardScaler()
X = scaler.fit_transform(location)
scaled_location = pd.DataFrame(X, index=location.index,
                                columns=location.columns)
scaled_location
```

	LAT	LON
0	-0.825199	0.843799
1	-1.020012	-0.349844
2	-0.481260	1.004899
3	0.201447	0.278538
4	-0.340753	1.038815
...
2113788	0.154899	0.094828
2113789	0.076457	-0.133161
2113790	-0.125252	0.678931
2113791	1.278952	-1.407825
2113792	-0.312307	-0.885900

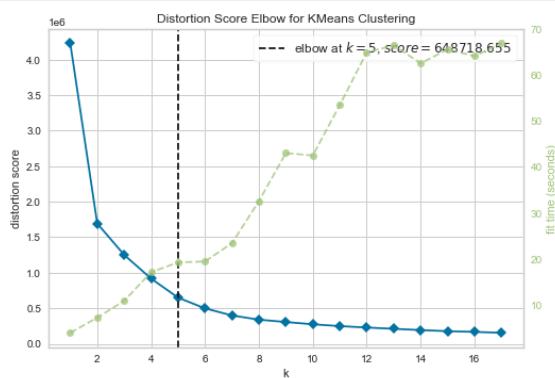
2113793 rows × 2 columns

- Βρίσκουμε με την χρήση του elbow method τον βέλτιστο αριθμό clusters για τα δεδομένα μας.

- Next up we will use the elbow method to find the best number of clusters.

```
In [102]: kmeans = KMeans(random_state=0)
visualizer = KElbowVisualizer(kmeans, k=(1, 8))

visualizer.fit(scaled_location)
visualizer.show()
```

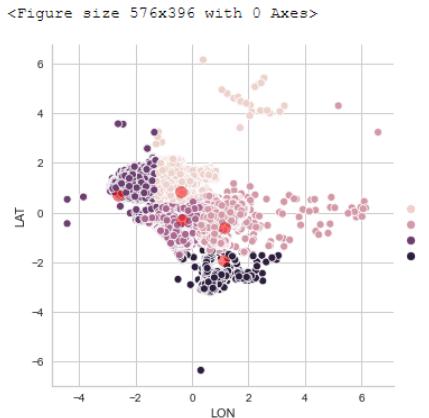


Σύμφωνα με το διάγραμμα, ο βέλτιστος αριθμός clusters για την συγκεκριμένη ανάλυση είναι 5. Αυτό σημαίνει ότι ο καλύτερος αριθμός clusters για τη δημιουργία ομάδων με στοιχεία που είναι πιο όμοια μεταξύ τους από τι με τα στοιχεία άλλων ομάδων, είναι ο αριθμός 5.

- Έτσι, υλοποιούμε τον αλγόριθμο kmeans clustering για αριθμό 5 clusters. Επίσης, οπτικοποιούμε τα δεδομένα και βλέπουμε τα clusters και τα cluster centroids.

```
In [138]: kmeans = KMeans(n_clusters=5, random_state=0).fit(scaled_location)
plt.figure()
sns.relplot(x='LON', y='LAT', hue=kmeans.labels_, data=scaled_location)
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],
            c='red', edgecolor='black', s=100, alpha=0.5)
```

Out[138]: <matplotlib.collections.PathCollection at 0x1af269dba90>



- Βέβαια στο συγκεκριμένο διάγραμμα δεν φαίνεται ο χάρτης του Los Angeles, οπότε τα οπτικοποιούμε πάνω στον χάρτη. Πρώτα, πρέπει να αποθηκεύσουμε βέβαια το cluster στο οποίο ανήκει το κάθε περιστατικό. Έπειτα εμφανίζουμε τα περιστατικά πάνω στο χάρτη του Los Angeles.

- Now we create a dataframe with the cluster each incident belongs to
- Then we merge the location dataframe with the kmeans labels in order to depict incidents clustering on the real map

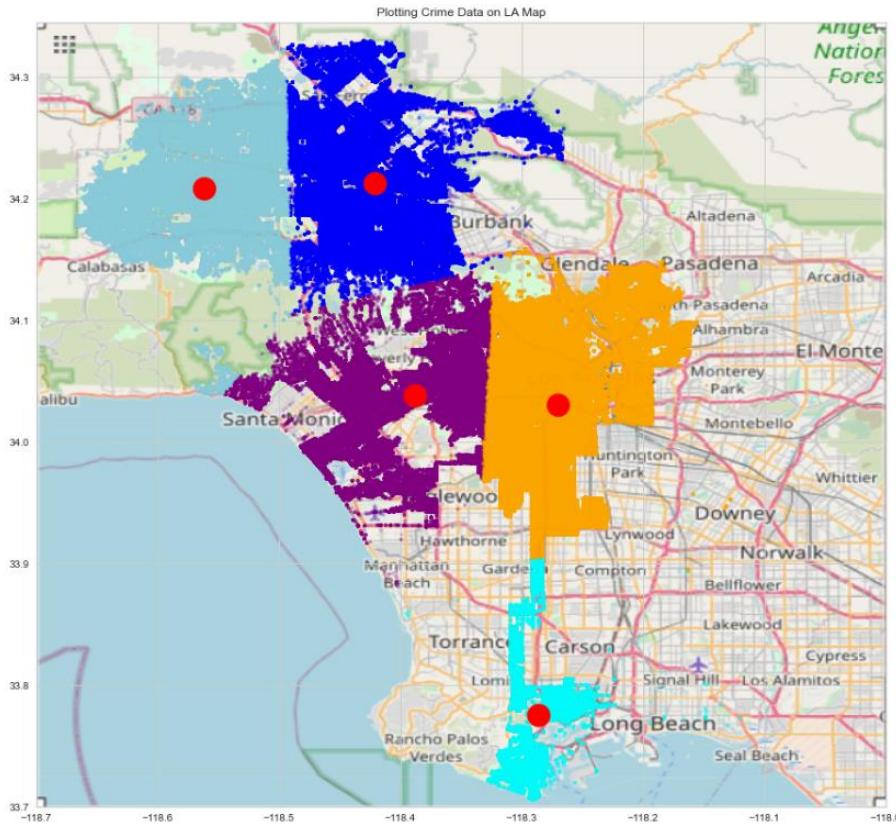
```
In [70]: location['label'] = pd.DataFrame(kmeans.labels_)
location
```

Out[70]:

	LAT	LON	label
0	33.9825	-118.2695	1
1	33.9599	-118.3962	2
2	34.0224	-118.2524	1
3	34.1016	-118.3295	2
4	34.0387	-118.2488	1
...
2113788	34.0962	-118.3490	2
2113789	34.0871	-118.3732	2
2113790	34.0637	-118.2870	1
2113791	34.2266	-118.5085	3
2113792	34.0420	-118.4531	2

2113793 rows × 3 columns

```
In [137]: BBox = ((-118.7000, -118, 33.7000, 34.3450)) # coordinates of map
ruh_m = plt.imread('lamap.png') #importing the map
fig,ax1 = plt.subplots(figsize = (16,14))
cmap = colors.ListedColormap(['blue','orange','purple','c','aqua']) # create new colormap
ax1.scatter(location.LON, location.LAT, zorder=1, alpha= 0.2, cmap=cmap,
            c=location.label, s=10, )
ax1.scatter(kmeans.cluster_centers_[:, 1], kmeans.cluster_centers_[:, 0], zorder=1, alpha= 1,color='red',
            s=500, ) #plotting the centroids
ax1.set_title('Plotting Crime Data on LA Map')
ax1.set_xlim(BBox[0],BBox[1])
ax1.set_ylim(BBox[2],BBox[3])
ax1.imshow(ruh_m, zorder=0, extent = BBox, aspect= 'equal')
```



Βλέπουμε στον χάρτη ότι τα διαφορετικά clusters φαίνονται με διαφορετικό χρώμα. Επίσης, οι κόκκινες τελείες αφορούν το centroid του κάθε cluster.

Με αυτό τον τρόπο, δημιουργούμε ομάδες (clusters) που περιέχουν περιστατικά που συνέβησαν σε κοντινές (όμοιες) περιοχές, ενώ περιστατικά διαφορετικών clusters συνέβησαν σε μακρινές (διαφορετικές) περιοχές. Αναλυτικά, με τον αλγόριθμο kmeans τα περιστατικά αντιστοιχούνται στο πλησιέστερο cluster (centroid). Μάλιστα, τα clusters ελαχιστοποιούν το μέσο όρο των τετραγώνων αποστάσεων (average of squared Euclidean distances) μεταξύ περιστατικών και centroid του συγκεκριμένου cluster. Παρόλα αυτά, δεν εγγυάται η επίτευξη ολικού ελάχιστου μέσω του k-means (Cambridge University Press, 2008)

Με βάση λοιπόν το σχήμα, θα μπορούσε κανείς να πει, ότι τα centroids είναι οι τοποθεσίες αστυνομικών τμημάτων σε περίπτωση που ήταν δυνατή μόνο η ύπαρξη 5 τμημάτων. Αφού τα centroids παράχθηκαν από τον αλγόριθμο kmeans, ελαχιστοποιείται ο μέσος όρος των τετραγώνων αποστάσεων μεταξύ περιστατικών ενός cluster με το centroid του. Παράλληλα, γνωρίζουμε ότι τα ιστορικά δεδομένα συνήθως επαναλαμβάνονται, οπότε περιοχές στις οποίες πραγματοποιήθηκαν πολλά εγκλήματα στο παρελθόν είναι πολύ πιθανό να έχουν την ίδια μοίρα και στον μέλλον.

Clustering on categorical variables

Σε αυτό το σημείο θα γίνει clustering των κατηγορικών μεταβλητών του είδους του εγκλήματος, του όπλου και των εγκαταστάσεων. Με αυτό τον τρόπο επιδιώκουμε να βγάλουμε clusters συγκεκριμένων τύπων εγκλήματος, συγκεκριμένου όπλου και συγκεκριμένης εγκατάστασης όπου συνέβη το έγκλημα. Έτσι, μπορούμε να βγάλουμε διάφορα συμπεράσματα και συνδυασμούς εγκλημάτων με όπλα και εγκαταστάσεις αλλά και αντίστροφα. Τέλος, αυτό θα γίνει με την χρήση του αλγόριθμου kmodes που χρησιμοποιείται για clustering κατηγορικών μεταβλητών.

- Αρχικά επιλέγουμε μόνο crime types που συνέβησαν πάνω από 5000 φορές μέσα στη δεκαετία για να μην επηρεαστούν τα αποτελέσματα από rare incidents

Clustering on categorical variables

- We will try to cluster by using categorical variables
- We will use kmodes to do that, an adaptation of kmeans where we can use categorical variables
- Kmodes defines clusters based on the number of matching categories between data points.
- We want to identify specific attributes combinations e.g. crime type, weapon used, victim sex that describe groups of crimes
- Only crime type, weapon used and premises are going to be used
- We are going to try clean the crimes from crime attributes that rarely appear

We pick crime types that have more than 5000 occurrences in the decade

```
In [87]: crime_types = la_crimes['CRM_DESC'].value_counts()[:35].index
crime_types = pd.DataFrame(crime_types, columns=['CRM_DESC'])
crime_types
```

	CRM_DESC
0	BATTERY - SIMPLE ASSAULT
1	BURGLARY FROM VEHICLE
2	VEHICLE - STOLEN
3	THEFT PLAIN - PETTY (\$950 & UNDER)
4	BURGLARY
5	THEFT OF IDENTITY
6	INTIMATE PARTNER - SIMPLE ASSAULT
7	VANDALISM - FELONY (\$400 & OVER, ALL CHURCH VA...
8	ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT
9	VANDALISM - MISDEAMEANOR (\$399 OR UNDER)
10	THEFT FROM MOTOR VEHICLE - PETTY (\$950 & UNDER)
11	ROBBERY
12	THEFT-GRAND (\$950.01 & OVER)EXCPT GUNS,FOWL,LI...

- Έπειτα, επιλέγουμε επίσης τα πιο συχνά όπλα και εγκαταστάσεις που χρησιμοποιούνται σε εγκλήματα.

We also pick weapon used types that have at least around 2500 occurrences in the decade

```
In [94]: weapon_types = la_crimes['WEAPON_DESC'].value_counts()[:23].index
weapon_types = pd.DataFrame(weapon_types, columns=['WEAPON_DESC'])
weapon_types
```

	WEAPON_DESC
0	unknown
1	STRONG-ARM (HANDS, FIST, FEET OR BODY FORCE)
2	VERBAL THREAT
3	UNKNOWN WEAPON/OTHER WEAPON
4	HAND GUN
5	SEMI-AUTOMATIC PISTOL
6	KNIFE WITH BLADE 6INCHES OR LESS
7	OTHER KNIFE
8	UNKNOWN FIREARM
9	VEHICLE
10	REVOLVER
11	BOTTLE

Lastly, premises types that have at least around 9000 occurrences in the decade are selected

```
In [99]: premises_types = la_crimes['PREMIS_DESC'].value_counts()[:22].index
premises_types = pd.DataFrame(premises_types, columns=['PREMIS_DESC'])
premises_types
```

	PREMIS_DESC
0	STREET
1	SINGLE FAMILY DWELLING
2	MULTI-UNIT DWELLING (APARTMENT, DUPLEX, ETC)
3	PARKING LOT
4	SIDEWALK
5	OTHER BUSINESS
6	VEHICLE, PASSENGER/TRUCK
7	DRIVEWAY
8	GARAGE/CARPORT
9	RESTAURANT/FAST FOOD
10	DEPARTMENT STORE
11	MARKET

➤ Κρατάμε τα περιστατικά που αφορούν τα πιο συχνά εγκλήματα, όπλα και εγκαταστάσεις.

- Now we merge all the data into a new dataframe which contains the most occurred crime types, weapons used and premises

```
In [101]: crimes_kmode = pd.merge (la_crimes, crime_types, on='CRM_DESC' )
crimes_kmode = pd.merge (crimes_kmode, weapon_types, on='WEAPON_DESC' )
crimes_kmode = pd.merge (crimes_kmode, premises_types, on='PREMIS_DESC' )
crimes_kmode
```

	DR_NO	DATE_OCC	AREA_NAME	CRM_DESC	Mocodes	VICT_AGE	VICT_SEX	VICT_DESC	PREMIS_DESC	WEAPON_DESC	STAT
0	1307355	2010-02-20 13:50:00	Newton	VIOLATION OF COURT ORDER	0913 1814 2000	48	M	H	SINGLE FAMILY DWELLING	unknown	A Ar
1	100205595	2010-01-30 20:25:00	Rampart	VIOLATION OF COURT ORDER	unknown	38	F	H	SINGLE FAMILY DWELLING	unknown	A Ol
2	100209830	2010-04-11 18:51:00	Rampart	VIOLATION OF COURT ORDER	unknown	25	F	H	SINGLE FAMILY DWELLING	unknown	Im C
3	100210217	2010-04-21 18:30:00	Rampart	VIOLATION OF COURT ORDER	0906 1816	0	M	H	SINGLE FAMILY DWELLING	unknown	Im C
4	100211380	2010-05-13 10:30:00	Rampart	VIOLATION OF COURT ORDER	1814 2000	29	F	H	SINGLE FAMILY DWELLING	unknown	A Ol
...

➤ Τώρα εκτελούμε τον αλγόριθμο kmodes για 10 clusters

- We execute the kmodes algorithm with 10 different clusters (groups)

```
In [102]: km = KModes(n_clusters=10, init='Huang', n_init=3, verbose=1)
clusters = km.fit_predict(crimes_kmode[['CRM_DESC','WEAPON_DESC','PREMIS_DESC']])
# Print the cluster centroids
print(km.cluster_centroids_)

Init: initializing centroids
Init: initializing clusters
Starting iterations...
Run 1, iteration: 1/100, moves: 280740, cost: 1844040.0
Run 1, iteration: 2/100, moves: 5288, cost: 1844040.0
Init: initializing centroids
Init: initializing clusters
Starting iterations...
Run 2, iteration: 1/100, moves: 209320, cost: 1882730.0
Run 2, iteration: 2/100, moves: 81725, cost: 1882730.0
Init: initializing centroids
Init: initializing clusters
Starting iterations...
Run 3, iteration: 1/100, moves: 145875, cost: 1831313.0
Best run was number 3
[['BURGLARY' 'unknown' 'SINGLE FAMILY DWELLING']
 ['TRESPASSING' 'unknown' 'DRIVEWAY']
 ['VANDALISM - MISDEAMEANOR ($399 OR UNDER)' 'unknown' 'OTHER BUSINESS']
 ['THEFT-GRAND ($950.01 & OVER)EXCEPT GUNS, FOWL, LIVESTK, PROD' 'unknown'
 'MULTI-UNIT DWELLING (APARTMENT, DUPLEX, ETC)']
 ['VEHICLE - STOLEN' 'unknown' 'STREET']
 ['BATTERY - SIMPLE ASSAULT'
 'STRONG-ARM (HANDS, FIST, FEET OR BODILY FORCE)' 'STREET']
 ['VANDALISM - MISDEAMEANOR ($399 OR UNDER)' 'unknown'
 'VEHICLE, PASSENGER/TRUCK']
 ['VANDALISM - FELONY ($400 & OVER, ALL CHURCH VANDALISMS)' 'unknown'
 'VEHICLE, PASSENGER/TRUCK']
 ['INTIMATE PARTNER - SIMPLE ASSAULT'
 'STRONG-ARM (HANDS, FIST, FEET OR BODILY FORCE)'
 'MULTI-UNIT DWELLING (APARTMENT, DUPLEX, ETC)']
 ['BURGLARY FROM VEHICLE' 'unknown' 'PARKING LOT']]
```

- Φαίνεται ότι στα περισσότερα clusters που βγάλαμε, ο τύπος του όπλου είναι άγνωστος. Για αυτό, αφαιρούμε τα περιστατικά με άγνωστο τύπο όπλου και ξανατρέχουμε τον αλγόριθμο προκειμένου να εξετάσουμε πάλι τα αποτελέσματα. Σε αυτό το σημείο είναι σημαντικό να πούμε ότι στα 1,4 εκατομμύρια περιστατικά από τα 2,1 εκατομμύρια το όπλο καταγράφηκε ως unknown. Αυτό μπορεί να σημαίνει πολλά. Όπως, ότι το θύμα δεν παρατήρησε το όπλο ή δεν θυμόταν το όπλο ή δεν γνώριζε το όπλο, πράγμα που συνιστά σημαντική λεπτομέρεια από μόνη της.

- We can see in most groups the weapon type is unknown.
- That's why we try to execute the algorithm without the rows with unknown weapon

```
In [106]: crimes_kmode = crimes_kmode.loc[crimes_kmode['WEAPON_DESC'] != 'unknown']

In [107]: km = KMeans(n_clusters=10, init='Huang', n_init=3, verbose=1)
clusters = km.fit_predict(crimes_kmode[['CRM_DESC','WEAPON_DESC','PREMIS_DESC']])

# Print the cluster centroids
print(km.cluster_centers_)

Init: initializing centroids
Init: initializing clusters
Starting iterations...
Run 1, iteration: 1/100, moves: 128277, cost: 548348.0
Run 1, iteration: 2/100, moves: 55227, cost: 543041.0
Run 1, iteration: 3/100, moves: 459, cost: 543041.0
Init: initializing centroids
Init: initializing clusters
Starting iterations...
Run 2, iteration: 1/100, moves: 169194, cost: 498624.0
Run 2, iteration: 2/100, moves: 22260, cost: 498624.0
Init: initializing centroids
Init: initializing clusters
Starting iterations...
Run 3, iteration: 1/100, moves: 130224, cost: 543223.0
Run 3, iteration: 2/100, moves: 802, cost: 543223.0
Best run was number 2
[('ROBBERY' 'STRONG-ARM (HANDS, FIST, FEET OR BODILY FORCE)' 'SIDEWALK')
 ('BATTERY - SIMPLE ASSAULT'
 'STRONG-ARM (HANDS, FIST, FEET OR BODILY FORCE)'
 'SINGLE FAMILY DWELLING')
 ('ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT' 'HAND GUN' 'STREET')
 ('INTIMATE PARTNER - SIMPLE ASSAULT'
 'STRONG-ARM (HANDS, FIST, FEET OR BODILY FORCE)'
 'SINGLE FAMILY DWELLING')
 ('CRIMINAL THREATS - NO WEAPON DISPLAYED' 'VERBAL THREAT'
 'SINGLE FAMILY DWELLING')
 ('INTIMATE PARTNER - SIMPLE ASSAULT'
 'STRONG-ARM (HANDS, FIST, FEET OR BODILY FORCE)'
 'MULTI-UNIT DWELLING (APARTMENT, DUPLEX, ETC)')
 ('ROBBERY' 'STRONG-ARM (HANDS, FIST, FEET OR BODILY FORCE)' 'STREET')
 ('BATTERY - SIMPLE ASSAULT'
 'STRONG-ARM (HANDS, FIST, FEET OR BODILY FORCE)' 'SIDEWALK')
 ('INTIMATE PARTNER - SIMPLE ASSAULT'
 'STRONG-ARM (HANDS, FIST, FEET OR BODILY FORCE)' 'STREET')
 ('BATTERY - SIMPLE ASSAULT'
 'STRONG-ARM (HANDS, FIST, FEET OR BODILY FORCE)' 'STREET')]
```

Στις παραπάνω εικόνες βλέπουμε επίσης τα clusters που υπολογίστηκαν.

Συμπεράσματα (στα αγγλικά για να μην αλλοιωθούν έννοιες και χαθεί πληροφορία):

- Usually, burglaries happen at a single-family dwelling with an unknown weapon. On the other side burglaries from vehicles take place at parking lots with unknown weapons
- Generally, trespassing concerns driveways of people
- Many grand thefts of 950.01 \$ and over happen at multi-unit dwellings as apartments, duplex etc. with different (unknown) weapons
- Usually, vehicles thefts happen at the street
- Typically, battery - simple assault takes place at the street, the sidewalk and single-family dwellings with the use of hands, fist, feet or bodily force as a weapon.
- Generally, vandalism both as a misdemeanor and felony happens at vehicles or trucks
- Most intimate partner simple assaults occur at a multi-unit dwelling as apartment or duplex or single-family dwellings with the use of hands, fist, feet or bodily force as a weapon.
- Typically, robberies take place at the sidewalk or street with the use of hands, fist, feet or bodily force as a weapon.
- Also, generally handguns are used for assaults with deadly weapon (aggravated assaults) which take place at the street

- Lastly, criminal threats with no weapon happen at single family dwellings with the use of verbal threats.

Σημείωση: Είναι πολύ σημαντικό να κατανοήσουμε πώς λειτουργεί ο αλγόριθμος kmode. Ο αλγόριθμος προσπαθεί να δημιουργήσει modes (ομάδες) που θα ταιριάζουν όσο το δυνατόν περισσότερα, παρόμοια περιστατικά σε κάθε συγκεκριμένη ομάδα. Αυτό σημαίνει ότι εάν ένα περιστατικό έχει δύο από τις τρεις μεταβλητές που διερευνούμε ίδιες με το centroid (cluster), πιθανότατα θα ανήκει στο συγκεκριμένο cluster (ομάδα). Ως αποτέλεσμα, τα cluster δεν μπορούν πάντα να περιγράφουν ομάδες με τις περισσότερες περιπτώσεις, αλλά σίγουρα συνδέουν διαφορετικούς τύπους εγκλημάτων με διαφορετικούς τύπους όπλων και διαφορετικούς τύπους εγκαταστάσεων που συμβαίνουν συνήθως.

Clustering & examining different districts criminality

Σε αυτό το σημείο θα γίνει clustering των district (περιοχών) με βάση το σύνολο των διαφορετικών τύπων εγκλημάτων που έγιναν σε κάθε district. Επίσης, θα ξαναχρησιμοποιηθεί ο αλγόριθμος kmeans.

- Ξεκινάμε με το να λάβουμε υπόψη μόνο τους πιο συνηθισμένους τύπους εγκλημάτων. Με άλλα λόγια, δεν λαμβάνουμε υπόψη τα σπάνια είδη εγκλημάτων

Clustering & examining different districts criminality

We start by clearing crime types that are very rare

```
In [70]: crime_types = la_crimes['CRM_DESC'].value_counts()[:100].index
crime_types = pd.DataFrame(crime_types, columns=['CRM_DESC'])
crime_types
```

```
Out[70]:
CRM_DESC
0          BATTERY - SIMPLE ASSAULT
1          BURGLARY FROM VEHICLE
2          VEHICLE - STOLEN
3          THEFT PLAIN - PETTY ($950 & UNDER)
4          BURGLARY
...
95         RECKLESS DRIVING
96         SHOPLIFTING - ATTEMPT
97  DEFRAUDING INNKEEPER/THEFT OF SERVICES, OVER $400
98  LEWD/LASCIVIOUS ACTS WITH CHILD
99  THEFT, COIN MACHINE - PETTY ($950 & UNDER)
```

100 rows × 1 columns

```
In [71]: la_crimes_clearedcrimes = pd.merge(la_crimes, crime_types, on='CRM_DESC')
la_crimes_clearedcrimes
```

	DR_NO	DATE_OCC	AREA_NAME	DISTR_NO	CRM_DESC	Mocodes	VICT_AGE	VICT_SEX	VICT_DESC	PREMIS_DESC	WEAPON_DESC
0	1307355	2010-02-20 13:50:00	Newton	1385	VIOLATION OF COURT ORDER	0913 1814 2000	48	M	H	SINGLE FAMILY DWELLING	unknown
1	90631215	2010-01-05 01:50:00	Hollywood	646	VIOLATION OF COURT ORDER	1100 0400 1402	47	F	W	STREET	HAND GUN
2	100104288	2010-01-08 10:00:00	Central	123	VIOLATION OF COURT ORDER	1501	31	F	B	GOVERNMENT FACILITY (FEDERAL, STATE, COUNTY & C...)	VERBAL THREAT
3	100105672	2010-02-01 18:40:00	Central	185	VIOLATION OF COURT ORDER	1501	34	M	W	OFFICE BUILDING/OFFICE	unknown
4	100105709	2010-02-02 16:00:00	Central	157	VIOLATION OF COURT ORDER	0601	0	M	B	OTHER RESIDENCE	unknown

- Σε αυτό το σημείο δημιουργούμε ένα pivot table που περιέχει για κάθε διαφορετικό district (κάθε district είναι μια γραμμή), το συνολικό αριθμό κάθε διαφορετικού τύπου εγκλήματος (κάθε τύπος εγκλήματος είναι μια στήλη) Παράλληλα αφαιρούμε τους τύπους εγκλημάτων που δεν σχετίζονται με το district αλλά προέρχονται από άλλους παράγοντες. Για παράδειγμα, letters, phone calls received as LEWD (underaged) & the intimate partner crimes are not connected directly to the place.

We create a pivot table with all the sum of all the different crimes for each district

```
In [72]: distr_crimes = la_crimes_clearedorimes.pivot_table(index="DISTR_NO", columns="CRM_DESC", values='DR_NO', aggfunc='count')
distr_crimes = distr_crimes.reset_index()
distr_crimes = distr_crimes.fillna(0)
```

- At this point we drop the rows that are not connected with the criminality of the particular district but stem from other factors. For example, the letters, phone calls received as LEWD (underaged) and the intimate partner crimes are not connected directly to the place

```
In [73]: distr_crimes = distr_crimes.drop(['INTIMATE PARTNER - SIMPLE ASSAULT', 'LETTERS, LEWD - TELEPHONE CALLS, LEWD',
'INTIMATE PARTNER - AGGRAVATED ASSAULT', 'CHILD NEGLECT (SEE 300 W.I.C.)',
'THREATENING PHONE CALLS/LETTERS'], axis=1)
distr_crimes
```

Out[73]:

CRM_DESC	DISTR_NO	ARSON	ASSAULT WITH DEADLY WEAPON ON POLICE OFFICER	ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT	ATTEMPTED ROBBERY	BATTERY - SIMPLE ASSAULT	BATTERY ON A FIREFIGHTER	BATTERY ON A POLICE (SIMPLE)	BATTERY WITH SEXUAL CONTACT	BIKE - STOLEN	THROWING OBJECT AT MOVING VEHICLE	TRESPASS
0	100	0.0	0.0	1.0	0.0	5.0	0.0	0.0	0.0	0.0	0.0	0.0
1	101	2.0	0.0	58.0	6.0	112.0	1.0	2.0	9.0	16.0	...	1.0
2	105	3.0	1.0	16.0	4.0	31.0	0.0	1.0	0.0	3.0	...	0.0
3	109	2.0	0.0	1.0	1.0	2.0	0.0	0.0	2.0	1.0	...	0.0
4	111	21.0	8.0	256.0	56.0	674.0	0.0	21.0	73.0	67.0	...	8.0
...
1298	2189	14.0	0.0	60.0	12.0	253.0	2.0	5.0	17.0	3.0	...	5.0
1299	2196	0.0	0.0	13.0	1.0	57.0	0.0	1.0	3.0	0.0	...	1.0
1300	2197	0.0	0.0	8.0	0.0	18.0	0.0	0.0	0.0	0.0	...	0.0
1301	2198	0.0	0.0	0.0	0.0	3.0	0.0	0.0	1.0	0.0	...	0.0
1302	2199	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0

1303 rows × 96 columns

- Αφαιρούμε επίσης την στήλη με τους κωδικούς των district αφού δεν βοηθά στο clustering

We create a new dataframe with the district codes and we drop the column because it is not useful for clustering

```
In [74]: distr_codes= pd.DataFrame()
distr_codes['DISTR_NO'] = distr_crimes['DISTR_NO'].copy()
distr_crimes = distr_crimes.drop(['DISTR_NO'], axis=1)
distr_codes
```

Out[74]:

	DISTR_NO
0	100
1	101
2	105
3	109
4	111
...	...
1298	2189
1299	2196
1300	2197
1301	2198
1302	2199

1303 rows × 1 columns

- Αφού χρησιμοποιούμε πάλι kmeans, πρέπει να αλλάξουμε τα δεδομένα με StandardScaler() για να έχουν όλες οι στήλες ίδια διακύμανση η οποία είναι ίση με 1. Εξάλλου, με την χρήση του StandardScaler όλες οι στήλες θα έχουν mean (μέσο όρο) ίσο με 0. Αυτό είναι απαραίτητο βήμα σε κάθε kmeans clustering ανάλυση.

- Now we will use cluster analysis to find groups with districts that are most similar to each other and most different to districts of other clusters when it comes to the types of crimes committed
- We will use StandardScaler() to transform the data so that all columns have equal variance. Necessary step in clustering

```
In [75]: scaler = StandardScaler()
X = scaler.fit_transform(distr_crimes)
scaled_distr_crimes = pd.DataFrame(X, index=distr_crimes.index,
                                     columns=distr_crimes.columns)
scaled_distr_crimes
```

CRM_DESC	ASSAULT WITH DEADLY WEAPON IN POLICE OFFICER	ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT	ATTEMPTED ROBBERY	BATTERY - SIMPLE ASSAULT	BATTERY ON A FIREFIGHTER	BATTERY POLICE (SIMPLE)	BATTERY WITH SEXUAL CONTACT	BIKE-STOLEN	BOAT-STOLEN	THROWING OBJECT AT MOVING VEHICLE	TRESPAS
0	-0.788017	-0.545349	-0.799151	-0.761967	-0.893136	-0.379811	-0.465930	-0.877134	-0.359013	-0.295999	...
1	-0.206598	-0.545349	-0.172559	-0.265131	-0.216489	1.142938	-0.216502	0.035973	0.162969	-0.295999	...
2	0.084112	-0.102081	-0.628282	-0.430743	-0.728717	-0.379811	-0.341216	-0.877134	-0.261142	-0.295999	...
3	-0.206598	-0.545349	-0.799151	-0.679161	-0.912108	-0.379811	-0.465930	-0.674221	-0.326389	-0.295999	...
4	5.316883	3.000048	2.105957	3.875163	3.337486	-0.379811	2.153080	0.529177	1.826789	-0.295999	3.970351
...
1298	3.281917	-0.545349	-0.126988	0.231704	0.675166	2.665688	0.157639	0.847623	-0.261142	-0.295999	2.201980
1299	-0.788017	-0.545349	-0.662440	-0.679161	-0.564298	-0.379811	-0.341216	-0.572765	-0.359013	-0.295999	...
1300	-0.788017	-0.545349	-0.719402	-0.761967	-0.810927	-0.379811	-0.465930	-0.877134	-0.359013	-0.295999	...
1301	-0.788017	-0.545349	-0.810543	-0.761967	-0.905784	-0.379811	-0.465930	-0.775677	-0.359013	-0.295999	...
1302	-0.788017	-0.545349	-0.810543	-0.761967	-0.924755	-0.379811	-0.465930	-0.877134	-0.359013	-0.295999	-0.745572

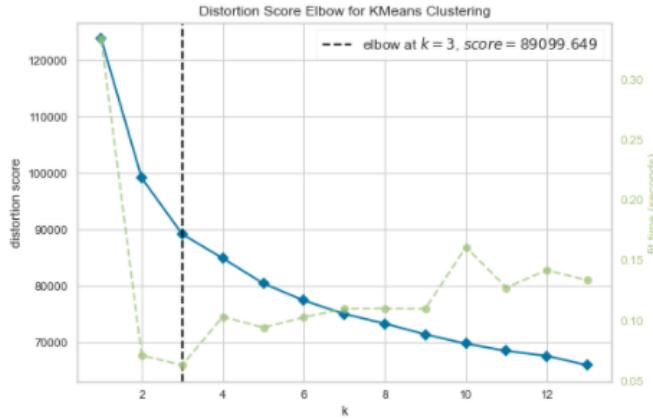
1303 rows × 95 columns

- Τώρα γίνεται χρήση της elbow method για να βρούμε τον βέλτιστο αριθμό clusters για τα δεδομένα μας. Φαίνεται πως είναι τα 3 clusters

Elbow method is used to find the best amount of clusters

```
In [414]: kmeans = KMeans(random_state=0)
visualizer = KElbowVisualizer(kmeans, k=(1,14))

visualizer.fit(scaled_distr_crimes)
visualizer.show()
```



```
Out[414]: <matplotlib.axes._subplots.AxesSubplot at 0x1af03cf6df0>
```

- Εκτελούμε τον kmeans αλγόριθμο με 3 clusters και ύστερα βλέπουμε τα centroids που υπολογίστηκαν.

```
In [425]: kmeans = KMeans(n_clusters=3, random_state=0, n_init=100).fit(scaled_distr_crimes)
print(kmeans.cluster_centers_)

[[-0.52986689 -0.36590593 -0.59160509 -0.55581315 -0.63506434 -0.25799087
-0.314747298 -0.58266697 -0.22241044 -0.11837339 -0.14750069 -0.65041124
-0.382155 -0.58285211 -0.52177121 -0.61284668 -0.59161117 -0.52409285
-0.5010974 -0.41062155 -0.58870352 -0.53639917 -0.42469259 -0.32281786
-0.3272882 -0.2275515 -0.21170436 -0.46093491 -0.70898282 -0.58241081
-0.50407029 -0.25272328 -0.17417272 -0.50364027 -0.4998027 -0.57435081
-0.2399982 -0.28868948 -0.01242943 -0.21488768 -0.594505 -0.33812233
-0.37885252 -0.19260187 -0.11479722 -0.27663263 -0.55319572 -0.49053366
-0.36508786 -0.38818987 -0.21746905 -0.47816986 -0.47072252 -0.47128574
-0.12899276 -0.3558874 -0.18065484 -0.17854667 -0.18975571 -0.40382953
-0.47035825 -0.58504711 -0.24859413 -0.39130227 -0.594945695 -0.08559798
-0.5427663 -0.39173198 -0.1890971 -0.184663201 -0.11129715 -0.39166268
-0.22513416 -0.45594522 -0.47864784 -0.56679809 -0.56987995 -0.76239044
-0.274761632 -0.6638175 -0.43534268 -0.47488379 -0.28660311 -0.44946736
-0.34364552 -0.38434211 -0.45008554 -0.35260384 -0.7705842 -0.72696165
-0.56595075 -0.70350156 -0.55819445 -0.6654727 -0.39676244]
[ 1.3660533 -1.33862139 2.4238946 2.28421659 2.40021718 0.80454965
1.66690696 1.812397798 0.66833798 0.1456653 0.74569278 2.27465587
0.971068202 0.92963344 1.16529304 1.40555802 1.39792305 1.39264508
1.22057644 1.79780088 1.69484422 1.16283958 1.64213437 0.57183706
0.9444362 0.84309683 0.54129048 1.96726041 2.26624988 1.5322475
1.15785898 0.83672356 0.49974786 1.32915287 0.69198033 1.17956996
0.48243792 1.55645309 0.0865024 0.79045006 1.2987817 1.32291762
1.19123558 0.78182528 0.93408269 0.40622445 1.36757041 2.022321
1.50573788 1.14872536 0.30614475 1.99730004 1.71466262 1.81521746
0.82771079 0.56465982 1.13658936 1.75117262 2.39194374 0.30159342
1.62355223 2.19986243 0.81650901 1.70847255 0.71208954 0.47500174
1.51571695 1.44344943 1.60104831 1.54874162 0.59118428 0.94295463
0.69060019 1.60104831 1.54874162 0.59118428 1.60717205
1.2758239 1.15125222 1.59638576 1.72248991 0.37946622 2.20792185
1.22732707 1.28499482 1.32703002 0.33980763 1.97185628 2.20319019
0.92283214 1.70428417 1.48250699 1.54590587 0.64976804]
[ 0.22742637 0.14757911 0.18942555 0.17664531 0.24156448 0.13044374
0.03117484 0.29548754 0.11723864 0.10329213 0.02044405 0.28218279
0.23503512 0.46295115 0.35132708 0.4045685 0.38044048 0.31079362
0.31818179 0.10960697 0.32437986 0.36782968 0.19453045 0.24561887
0.17992284 0.08976362 0.12957167 0.13266701 0.34799874 0.34828777
0.33332484 0.11857176 0.09628979 0.3039765 0.37273473 0.40627316
0.17174664 0.02164293 0.0312424 0.08585298 0.33999509 0.12009777
0.18750428 0.06305106 -0.05110996 0.2285494 0.32554477 0.15465635
0.11501794 0.20799268 0.18044654 0.14696975 0.19125865 0.17282803
-0.01539898 0.28323923 -0.01726733 -0.02518523 0.15041688 0.09858879
0.20912109 0.22249 0.11787431 0.09725838 0.14926434 0.03671342
0.30795002 0.156035 0.07310927 0.06752515 0.03203603 0.156823892
0.11600706 0.19987289 0.23138738 0.29017703 0.44620328 0.53143762
0.05937933 0.50973806 0.17487629 0.1943394 0.20946418 0.07449392
0.14423619 0.17796303 0.24207758 0.32224241 0.47132394 0.37960761
0.4489971 0.44846629 0.33115926 0.43679247 0.31193259]]
```

Ελέγχοντας τα centroids, μπορούμε να βγάλουμε συμπεράσματα για την ασφάλεια του district:

Αναλυτικά, το πρώτο centroid έχει τιμές κάτω του 0 για όλες τις στήλες κατά αρκετό ποσό (δηλαδή στις περισσότερες φορές έχει τιμές μικρότερες του -0.2). Γνωρίζοντας ότι μετά τη χρήση του StandardScaler() όλες οι στήλες εγκλημάτων έχουν μέση τιμή ίση με 0, συνεπάγεται ότι αυτό το centroid έχει τιμές μικρότερες του μέσου όρου για κάθε διαφορετικό τύπο εγκλήματος. Άρα αυτό το centroid περιγράφει τις ασφαλέστερες περιοχές, δηλαδή αυτές με τους λιγότερους τύπους εγκλημάτων και την χαμηλή εγκληματικότητα.

Το δεύτερο centroid έχει τιμές πάνω από 1 στις περισσότερες στήλες (τύπους εγκλημάτων). Έτσι, αυτό το centroid έχει τιμές πολύ μεγαλύτερες του μέσου όρου για κάθε διαφορετικό τύπο εγκλήματος. Συμπερασματικά, αυτό το centroid περιγράφει τις επικίνδυνες περιοχές, δηλαδή αυτές με τους περισσότερους τύπους εγκλημάτων.

Τέλος, το τρίτο centroid έχει τιμές ελαφρώς πάνω (μερικές φορές κάτω) από 0 για όλες τις στήλες. Έτσι, αυτό το centroid έχει τιμές που κυμαίνονται στο μέσο όρο για κάθε διαφορετικό τύπο εγκλήματος. Οπότε αυτό το centroid περιγράφει τις μέσης ασφάλειας περιοχές, δηλαδή αυτές με την μέση εγκληματικότητα.

- Σε αυτό το σημείο αποθηκεύουμε τον αριθμό cluster της κάθε district
- Έπειτα, αλλάζουμε την σειρά των clusters μεταξύ του δεύτερου και του τρίτου cluster, προκειμένου να υπάρχει μια συνέχεια όσον αφορά την ασφάλεια δηλαδή

στο cluster 0 να είναι οι ασφαλείς περιοχές, στο cluster 1 να είναι οι μέσης ασφάλειας περιοχές και στο cluster 2 να είναι οι επικίνδυνες περιοχές.
Τέλος, φτιάχνουμε ένα νέο dataframe που για κάθε περιστατικό έχει τις γεωγραφικές συντεταγμένες του και τον αριθμό cluster στον οποίο αντιστοιχεί το district όπου συνέβη.

```
We get for each district its label in order to depict it on the map

In [426]: distr_codes['label'] = pd.DataFrame(kmeans.labels_)
distr_codes

Out[426]:
   DISTR_NO  label
0          100      0
1          101      2
2          105      0
3          109      0
4          111      1
...
1288     2189      2
1299     2196      0
1300     2197      0
1301     2198      0
1302     2199      0
1303 rows × 2 columns

• It would be more suiting to swap values of labels for label 1 and 0 (because label 0 is medium safe and 1 is safe district) in order to have a continuous extent.

In [427]: indexes_2 = distr_codes.loc[distr_codes['label'] == 2].index
indexes_1 = distr_codes.loc[distr_codes['label'] == 1].index
distr_codes.loc[indexes_2, 'label'] = 1
distr_codes.loc[indexes_1, 'label'] = 2

Out[427]:
New dataframe that includes labels, district and longitude and latitude for every case

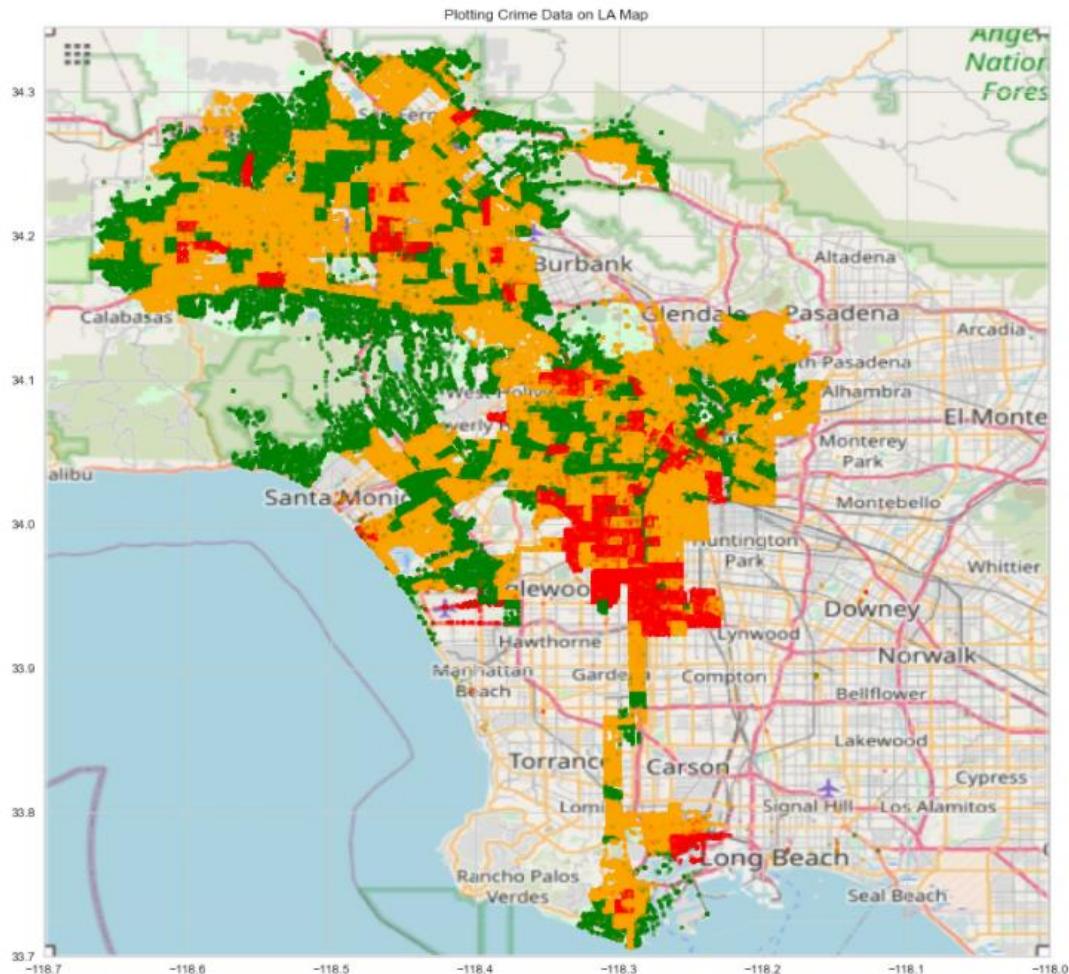
In [428]: crime_data_district = pd.merge(distr_codes, la_crimes[['DISTR_NO','LAT','LON']], on='DISTR_NO')
crime_data_district

Out[428]:
   DISTR_NO  label      LAT      LON
0          100      0  34.0428 -118.2461
1          100      0  34.0428 -118.2461
2          100      0  34.0423 -118.2452
3          100      0  34.0400 -118.2509
4          100      0  34.0407 -118.2680
...
2114694    2198      0  34.1483 -118.0034
2114695    2198      0  34.1487 -118.0054
```

➤ Τώρα μπορούμε να οπτικοποιήσουμε τα δεδομένα όλων των περιστατικών

- We will use a map in order to showcase the results

```
In [423]: BBox = ((-118.7000, -118, 33.7000, 34.3450)) # coordinates of map
ruh_m = plt.imread('litmap.png') #importing the map
fig, (ax1,ax2) = plt.subplots(1,2,figsize = (32,28))
cmap = colors.ListedColormap(['green','orange','red']) # create new colormap
ax1.scatter(crime_data_district.LON, crime_data_district.LAT, zorder=1, alpha= 0.2, cmap=cmap,
            c=crime_data_district.label, s=10, )
ax1.set_title('Plotting Crime Data on LA Map')
ax1.set_xlim(BBox[0],BBox[1])
ax1.set_ylim(BBox[2],BBox[3])
ax1.imshow(ruh_m, zorder=0, extent = BBox, aspect= 'equal')
ax2.imshow(ruh_m, zorder=0, extent = BBox, aspect= 'equal')
```



Στον χάρτη μπορούμε να δούμε τρία διαφορετικά χρώματα σημείων. Κάθε σημείο είναι ένα αναφερόμενο περιστατικό εγκλήματος από το 2010 έως το 2019. Πρώτον, ομαδοποιήσαμε ανά district, λαμβάνοντας το άθροισμα των μετρήσεων για κάθε διαφορετικό είδος εγκλήματος που διαπράχθηκε κάθε district. Στη συνέχεια ομαδοποιήσαμε τις περιοχές σε 3 διαφορετικά clusters. Αφού ανακαλύψουμε ότι κάθε διαφορετικό cluster ορίζει την ασφάλεια της περιοχής λόγω των τιμών των centroids (ένα centroid έχει αρνητικές τιμές μετά το normalization, ένα centroid έχει τιμές περίπου 0 και το άλλο centroid τιμές περίπου 1), μπορούμε να βγάλουμε συμπεράσματα σχετικά με την ασφάλεια των district.

- Τα πράσινα σημεία του χάρτη αφορούν district με λιγότερα εγκλήματα από ότι συνήθως. Αυτές είναι οι ασφαλέστερες ζώνες. Αυτό συμβαίνει επειδή αυτές οι περιοχές ανήκουν στο cluster με τις αρνητικές τιμές του centroid. Μερικά παραδείγματα είναι: Bel Air, Beverly Hills, West Hollywood, Brentwood, Mulholland Dr / Sepulveda Blvd και Palisades Dr / Ave De Santa Ynez

- Τα πορτοκαλί σημεία του χάρτη αφορούν district με μέσο όρο εγκλημάτων. Αυτές δεν είναι ούτε οι ασφαλέστερες ούτε οι πιο επικίνδυνες ζώνες. Αυτό εξηγείται από το γεγονός ότι αυτές οι περιοχές ανήκουν στο cluster με τις τιμές του centroid που είναι περίπου 0. Μερικά παραδείγματα είναι: Harbour Blvd, Forest Lawn Dr, Dolanco Junction και Tampa Ave
- Τα κόκκινα σημεία των χαρτών αφορούν district με περισσότερα εγκλήματα από ότι συνήθως και θεωρούνται επικίνδυνες ζώνες. Αυτό συμβαίνει επειδή αυτές οι περιοχές ανήκουν στο cluster με τις υψηλές (θετικές) τιμές του centroid. Παραδείγματα είναι Downtown, South Park, Central city και Fashion District.

Σημείωση: οι περιοχές που δεν έχουν χρώμα είναι είτε πολύ ασφαλείς (δεν έχουν διαπραχθεί εγκλήματα) είτε δεν έχουν αναφερθεί εγκλήματα δεδομένων ή λείπουν τα περιστατικά εγκλημάτων. Για παράδειγμα, για το Marvin Braude Mulholland Gateway Park (που βρίσκεται στο βορειοδυτικό Los Angeles, καταπράσινο δάσος) μπορούμε να υποθέσουμε ότι δεν υπάρχουν εγκλήματα που διαπράπτονται εκεί επειδή οι άνθρωποι δεν ζουν εκεί. Ωστόσο, για την περιοχή του Torrance μπορούμε να υποθέσουμε ότι τα εγκλήματα για αυτήν την περιοχή λείπουν αφού δεν υπάρχει κανένα περιστατικό.

Prediction

Τώρα θα προσπαθήσουμε να προβλέψουμε το είδος (τύπο) του εγκλήματος γνωρίζοντας όλα τα άλλα δεδομένα για κάθε έγκλημα (βέβαια όχι το primary crime code). Λεπτομερώς, οι στήλες (διαστάσεις) που θα χρησιμοποιηθούν για την πρόβλεψη του είδους του εγκλήματος είναι:

- Ήρα του περιστατικού
- Μέρα της εβδομάδας του περιστατικού
- AREA
- District
- Part 1 or 2 type
- Mocodes που είναι δραστηριότητες που σχετίζονται με τον ύποπτο για τη διάπραξη του εγκλήματος
- Victim age
- Victim sex
- Victim descent
- Premises (εγκαταστάσεις όπου συνέβη το περιστατικό)
- Weapon used
- Status
- Extra crimes committed (λιγότερο σοβαρά)
- Location name
- Longitude and latitude (γεωγραφικές συντεταγμένες)

Στο συγκεκριμένο μοντέλο πρόβλεψης θα χρησιμοποιήσουμε τα decision trees και τις επεκτάσεις τους. Τα decision trees συνιστούν machine learning algorithm τύπου επιβλεπόμενης μάθησης (supervised learning). Πολύ σημαντικό πλεονέκτημα των decision trees είναι η ευκολία ερμηνείας και εξήγησης του τρόπου λειτουργίας τους, σε αντίθεση με διάφορους άλλους αλγόριθμους μηχανικής μάθησης όπως τα νευρωνικά δίκτυα.

- Ξεκινάμε εισάγοντας τις απαραίτητες βιβλιοθήκες για την ανάλυση.
Έστερα από αυτό, επιλέγουμε μόνο τα δεδομένα 2 μηνών (Απρίλιου και Μαΐου 2019) λόγω περιορισμών RAM. Συγκεκριμένα, η ύπαρξη τόσων πολλών και διαφορετικών categorical variables απαιτεί την δημιουργία dummy variables σαν νέες στήλες η οποίες δεσμεύουν πολύ μνήμη. Βέβαια, αν είχαμε στη διάθεση μας περισσότερη RAM θα ήταν δυνατή η επεκτασιμότητα του μοντέλου πρόβλεψης για όλη την δεκαετία.

- Importing the libraries

```
In [64]: from sklearn.model_selection import train_test_split
import random
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn import tree
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.ensemble import AdaBoostClassifier

from sklearn.metrics import classification_report
```

- We only select the data of two months (2019 April and May) because of memory RAM restrictions. In detail, because of the fact that most of the columns are categorical variables, it will result in a huge amount of columns after we create dummy variables to depict our categorical data.

```
In [65]: la_crimes_new = la_crimes.loc[(la_crimes['DATE_OCC'].dt.to_period('M') == '2019-04') |
                                  (la_crimes['DATE_OCC'].dt.to_period('M') == '2019-05')].copy()
```

out[65]:	DR_NO	DATE_OCC	AREA_NAME	DISTR_NO	PART1_2	CRM_CD	CRM_DESC	Mocodes	VICT_AGE	VICT_SEX	VICT_DESC	PREMIS_DESC
	1847402	190408859 2019-04-21 18:30:00	Hollenbeck	449	1	510	VEHICLE - STOLEN	unknown	0	X	X	STREET
	1848938	190111434 2019-04-17 04:00:00	Central	147	1	510	VEHICLE - STOLEN	unknown	0	X	X	STREET
	1848966	190508840 2019-04-21 20:00:00	Harbor	566	1	510	VEHICLE - STOLEN	unknown	0	X	X	STREET
	1898385	191715448 2019-05-29 07:30:00	Devonshire	1798	2	901	VIOLATION OF RESTRAINING ORDER	1501 2038 1906 2000	54	F	W	SINGLE FAMILY DWELLING
	1898387	190715605 2019-05-26 10:30:00	Wilshire	729	2	956	LETTERS, LEWD - TELEPHONE CALLS, LEWD	1906	47	M	O	SINGLE FAMILY DWELLING

- Τώρα δημιουργούμε 2 νέες στήλες από την στήλη DATE_OCC: μια που περιέχει την ημέρα της εβδομάδας και μια που έχει την ώρα που συνέβη το περιστατικό εγκλήματος. Επιβεβαιώσαμε ότι αυτά τα μέτρα χρόνο αυξάνουν την ακρίβεια των προβλέψεων.

- We will also create new categorical columns that will contain the month and hour of the incident
- We make those columns from the column DATE_OCC

```
In [66]: la_crimes_new['WEEKDAY'] = la_crimes_new['DATE_OCC'].dt.dayofweek.astype(str).copy()
la_crimes_new['HOUR'] = la_crimes_new['DATE_OCC'].astype(str).str[11:13].copy()
```

- Δημιουργούμε ένα DataFrame που έχει μόνο τις στήλες που αναφέρθηκαν στην αρχή

Now we create a dataset that only contains the columns that are useful. Those columns are dropped:

- DR_NO does not help as it is a unique number
- DATE_OCC also is cut because we already have the hour and day in different columns

```
In [67]: la_crimes_dum = la_crimes_new[['LAT', 'LON', 'LOCATION', 'CRM_CD2', 'CRM_CD3', 'CRM_CD4', 'VICT_AGE', 'Mocodes', 'WEEKDAY', 'AREA_NAME', 'DISTR_NO', 'CRM_DESC', 'PREMIS_DESC', 'HOUR', 'WEAPON_DESC', 'VICT_DESC', 'VICT_SEX', 'PART1_2']].copy()
la_crimes_dum
```

	LAT	LON	LOCATION	CRM_CD2	CRM_CD3	CRM_CD4	VICT_AGE	Mocodes	WEEKDAY	AREA_NAME	DISTR_NO	CRM_DESC	PREMIS_DESC
1847402	34.0516	-118.1982	DOBISON	-1	-1	-1	0	unknown	6	Hollenbeck	449	VEHICLE - STOLEN	
1848938	34.0453	-118.2443	400 S SAN PEDRO ST	-1	-1	-1	0	unknown	2	Central	147	VEHICLE - STOLEN	
1848966	33.7360	-118.2857	900 S MESA ST	-1	-1	-1	0	unknown	6	Harbor	566	VEHICLE - STOLEN	
1898385	34.2320	-118.4742	8900 AQUEDUCT AV	-1	-1	-1	54	1501 2038 1906 2000	2	Devonshire	1798	VIOLATION OF RESTRAINING ORDER	DW
1898387	34.0711	-118.3248	200 S LUCERNE BL	-1	-1	-1	47	1906	6	Wilshire	729	LETTERS, LEWD - TELEPHONE CALLS, LEWD	DW
...
2114670	34.2376	-118.5947	9200 ETON	-1	-1	-1	0	unknown	5	Devonshire	1761	VEHICLE - STOLEN	

- Τώρα φτιάχνουμε τα dummy variables για την στήλη Mocodes. Το πρόβλημα για την συγκεκριμένη στήλη είναι κάποια περιστατικά μπορεί να έχουν πολλούς διαφορετικούς κωδικούς mpcodes. Έτσι, αν χρησιμοποιούσαμε την απλή εντολή ‘get_dummies()’ θα δημιουργούνταν και dummy variables που θα είχαν δύο ή τρεις ή παραπάνω mpcodes μαζί (π.χ. dummy variable Mocode_107_3030), πράγμα που θα ήταν λάθος. Θα χάναμε την πληροφορία εγκλημάτων που είχαν παρόμοια mpcodes και θα δεσμεύαμε και παραπάνω μνήμη.
- Για αυτό διαχωρίζουμε τα mpcodes σε κάθε περιστατικό και έπειτα φτιάχνουμε τα dummies. Με αυτό τον τρόπο κάθε mocode μπορεί να αντιστοιχεί σε πολλά διαφορετικά περιστατικά και κάθε dummy περιέχει μόνο έναν μοναδικό Mocode.

Now we create the dummies for the column `Mocodes`. The problem for this column is that one particular incident may have more than one different mpcodes. As a result 'get_dummies()' would create new dummies for multiple (pairs, triplets of) mpcodes which would be a mistake. We would miss crimes that contain similar mpcodes and the dataframe would get even bigger.

- We separate the mpcodes from each other and then execute the dummies
- Each mocode may correspond to multiple incidents

```
In [68]: mpcodes_dummies= la_crimes_dum['Mocodes'].str.join(sep='').str.get_dummies(sep=' ')
mpcodes_dummies
```

	0100	0101	0102	0104	0105	0107	0109	0110	0112	0113	...	3030	3034	3037	3101	3401	3701	4018	4026	9999	unknown
1847402	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1
1848938	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1
1848966	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1
1898385	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1898387	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...	
2114670	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1
2114672	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2114676	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2114679	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2114685	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

36383 rows x 518 columns

- Έπειτα ενώνουμε τα dummies με τις άλλες στήλες μας

Next up we join the data of the mocode dummies and `la_crimes_dum`

```
In [69]: la_crimes_dum = pd.merge(la_crimes_dum,mocodes_dummies, left_on=la_crimes_dum.index, right_on=mocodes_dummies.index )
la_crimes_dum = la_crimes_dum.drop(['key_0'], axis=1) # dropping the new column of the index(keys)
la_crimes_dum
```

	LAT	LON	LOCATION	CRM_CD2	CRM_CD3	CRM_CD4	VICT_AGE	Mocodes	WEEKDAY	AREA_NAME	...	3030	3034	3037	3101	3401
0	34.0516	-118.1982	DOBINSON	-1	-1	-1	0	unknown	6	Hollenbeck	...	0	0	0	0	0
1	34.0453	-118.2443	400 S SAN PEDRO ST	-1	-1	-1	0	unknown	2	Central	...	0	0	0	0	0
2	33.7360	-118.2857	900 S MESA ST	-1	-1	-1	0	unknown	6	Harbor	...	0	0	0	0	0
3	34.2320	-118.4742	8900 AQUEDUCT AV	-1	-1	-1	54	1501 2038 1906 2000	2	Devonshire	...	0	0	0	0	0
4	34.0711	-118.3248	200 S LUCERNE BL	-1	-1	-1	47	1906	6	Wilshire	...	0	0	0	0	0
...
36378	34.2376	-118.5947	9200 ETON AV	-1	-1	-1	0	unknown	5	Devonshire	...	0	0	0	0	0

- Τώρα θα προσπαθήσουμε να κάνουμε ότι κάναμε για τα mpcodes, στα Crm Cd 2, Crm Cd 3 και Crm Cd 4 (δευτερεύοντα εγκλήματα).

Έτσι, βάζουμε όλους τους κωδικούς δευτερευόντων εγκλημάτων στην ίδια στήλη και διαγράφουμε τις άλλες δυο.

- Now we will do the same with the columns `CRM_CD2` `CRM_CD3` `CRM_CD4`
- At first we join the string into one column `CRM_CD2`
- Drop the other two columns

```
In [70]: la_crimes_dum['CRM_CD2'] = la_crimes_dum['CRM_CD2'].astype(str)+ ' ' +la_crimes_dum['CRM_CD3'].astype(str) + ' '+ \
+la_crimes_dum['CRM_CD4'].astype(str)
la_crimes_dum = la_crimes_dum.drop(['CRM_CD3','CRM_CD4'], axis=1)
la_crimes_dum
```

	LAT	LON	LOCATION	CRM_CD2	VICT_AGE	Mocodes	WEEKDAY	AREA_NAME	DISTR_NO	CRM_DESC	...	3030	3034	3037	3101	3401
0	34.0516	-118.1982	DOBINSON	-1-1-1	0	unknown	6	Hollenbeck	449	VEHICLE - STOLEN	...	0	0	0	0	0
1	34.0453	-118.2443	400 S SAN PEDRO ST	-1-1-1	0	unknown	2	Central	147	VEHICLE - STOLEN	...	0	0	0	0	0
2	33.7360	-118.2857	900 S MESA ST	-1-1-1	0	unknown	6	Harbor	566	VEHICLE - STOLEN	...	0	0	0	0	0
3	34.2320	-118.4742	8900 AQUEDUCT AV	-1-1-1	54	1501 2038 1906 2000	2	Devonshire	1798	VIOLATION OF RESTRAINING ORDER	...	0	0	0	0	0
4	34.0711	-118.3248	200 S LUCERNE BL	-1-1-1	47	1906	6	Wilshire	729	LETTERS, LEWD - TELEPHONE CALLS, LEWD	...	0	0	0	0	0
...

- Διαχωρίζουμε τα Crm codes το ένα από το άλλο και ύστερα φτιάχνουμε τα dummies. Τώρα κάθε κωδικός δευτερεύοντος εγκλήματος μπορεί να αντιστοιχεί σε πολλά διαφορετικά incident και κάθε dummy αντιστοιχεί σε ένα ξεχωριστό Crm code.

- We separate the extra crime codes() that are located in CRM_CD2 from each other and then produce the dummies
- Each extra crime code may correspond to multiple incidents

```
In [71]: crime_codes_dummies = la_crimes_dum['CRM_CD2'].str.join(sep='').str.get_dummies(sep=' ')
crime_codes_dummies = crime_codes_dummies.drop(['-1'], axis=1) # dropping -1 column as it concerns incidents with no ex:
crime_codes_dummies
```

	210	231	235	236	320	330	341	343	345	350	...	930	933	940	946	956	990	993	997	998	999
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...
36378	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
36379	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
36380	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
36381	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
36382	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
36383	36383 rows × 68 columns																				

Σημείωση: Τα dummies με 3 αριθμητικά ψηφία αφορούν crime codes ενώ τα dummies με 4 ψηφία αφορούν Mocodes.

➤ Ενώνουμε πάλι τα dummies με τις άλλες στήλες μας

Next up we join the data of the crime codes dummies and la_crimes_dum

```
In [72]: la_crimes_dum = pd.merge(la_crimes_dum, crime_codes_dummies, left_on=la_crimes_dum.index,
                               right_on=crime_codes_dummies.index)
la_crimes_dum = la_crimes_dum.drop(['key_0'], axis=1) # dropping the new column of the index(keys)
la_crimes_dum
```

	LAT	LON	LOCATION	CRM_CD2	VICT_AGE	Mocodes	WEEKDAY	AREA_NAME	DISTR_NO	CRM_DESC	...	930	933	940	946	956
0	34.0516	-118.1982	DOBISON	-1-1-1	0	unknown	6	Hollenbeck	449	VEHICLE - STOLEN	...	0	0	0	0	0
1	34.0453	-118.2443	400 S SAN PEDRO ST	-1-1-1	0	unknown	2	Central	147	VEHICLE - STOLEN	...	0	0	0	0	0
2	33.7360	-118.2857	900 S MESA ST	-1-1-1	0	unknown	6	Harbor	566	VEHICLE - STOLEN	...	0	0	0	0	0
3	34.2320	-118.4742	8900 AQUEDUCT AV	-1-1-1	54	1501 2038 1906 2000	2	Devonshire	1798	VIOLATION OF RESTRAINING ORDER	...	0	0	0	0	0
4	34.0711	-118.3248	200 S LUCERNE BL	-1-1-1	47	1906	6	Wilshire	729	LETTERS, LEWD - TELEPHONE CALLS, LEWD	...	0	0	0	0	0
...

➤ Δημιουργούμε τα dummy variables για τις υπόλοιπες στήλες και διαγράφουμε τις στήλες Mocodes και CRM_CD2. Επίσης δημιουργούμε και dummies για το Victim Age αφού τα missing values έχουν την τιμή 0 και θα επηρέαζαν τις προβλέψεις αν εξετάζαμε την στήλη σαν συνεχής μεταβλητή.

- Now we can create the dummies for all the other variables.
- We get dummies for all the columns except for LAT, LON and CRM_DESC. Note: we get dummies for the column victim age as well, because the missing values are coded as 0. If we used the age as a continuous variable, it could produce wrong results.

```
In [73]: la_crimes_dum = pd.get_dummies(la_crimes_dum, columns=['LOCATION','VICT_AGE',
                                                               'AREA_NAME','DISTR_NO', 'PREMIS_DESC', 'WEAPON_DESC',
                                                               'VICI_DESC', 'VICT_SEX', 'PART1_2','WEEKDAY','HOUR'])
la_crimes_dum = la_crimes_dum.drop(['Mocodes','CRM_CD2'], axis=1)
la_crimes_dum
```

	LAT	LON	CRM_DESC	0100	0101	0102	0104	0105	0107	0109	...	HOUR_14	HOUR_15	HOUR_16	HOUR_17	HOUR_18	HOUR_19	H...
0	34.0516	-118.1982	VEHICLE-STOLEN	0	0	0	0	0	0	0	...	0	0	0	0	1	0	
1	34.0453	-118.2443	VEHICLE-STOLEN	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
2	33.7360	-118.2857	VEHICLE-STOLEN	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
3	34.2320	-118.4742	VIOLATION OF RESTRAINING ORDER	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
4	34.0711	-118.3248	LETTERS, LEWD-TELEPHONE CALLS, LEWD	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
...	

- Τελευταίο βήμα είναι να ανακατεύσουμε τα δεδομένα μας, να διαχωρίσουμε την στήλη που θα προβλέψουμε και να διαχωρίσουμε τα δεδομένα σε δεδομένα εκπαίδευσης και δεδομένα testing. Συγκεκριμένα, το 20% των δεδομένων χρησιμοποιείται για σκοπούς testing.

- Firstly, we shuffle the data
- We separate the target column from the other columns
- Now we split the data into training and testing. Specifically, we use 20% for testing purposes
- The target column is CRM_DESC
- Kfold cross validation is not used due to memory RAM restrictions

```
In [74]: la_crimes_dum = la_crimes_dum.sample(frac = 1, random_state=9) #shuffling data
target = la_crimes_dum['CRM_DESC'].copy() # target column separated
la_crimes_dum = la_crimes_dum.drop(['CRM_DESC'], axis=1) #deleting target column from data
X_train, X_test, y_train, y_test = train_test_split(la_crimes_dum, target, test_size=0.2, random_state=7)
```

Decision Trees

Το πρώτο μοντέλο πρόβλεψης που δημιουργήθηκε συνιστά ένα decision tree. Στο decision tree όπως και στα υπόλοιπα μοντέλα πρόβλεψης χρησιμοποιούμε το κριτήριο gini αφού δίνει καλύτερα αποτελέσματα από το κριτήριο entropy.

- Εκπαιδεύουμε το μοντέλο με τα train data και έπειτα προβλέπουμε το είδος εγκλήματος για τα test data

Decision Tree

First predictive model that is going to be used is a decision tree

- Firstly a classifier is created. For the decision tree and all the classifiers the criterion `gini` is used because it produces better accuracy compared to `entropy`
- Then we train it with the train data
- Lastly, we predict the test data

```
In [75]: clf_gini = tree.DecisionTreeClassifier(criterion='gini')
clf_gini = clf_gini.fit(X_train, y_train)
gini_predict = clf_gini.predict(X_test)
```

- Τώρα μπορούμε να δούμε πόσο καλά πήγε το μοντέλο πρόβλεψης μας στα δεδομένα testing

Now we can see how our decision tree performed

- 67% total accuracy. Accuracy is the proportion of instances correctly classified by the classifier.
- Weighted average of precision is 66%. Precision is the ratio of correctly reported positives over all reported positives
- Weighted average of recall is 67%. Recall is the ratio of correctly reported positives over all actual positives
- Weighted average of f1-score is 66%. F1-score is the harmonic mean of the precision and the recall.

```
In [78]: print(classification_report(y_test, gini_predict, zero_division=1))
```

	precision	recall	f1-score	support
ARSON	0.43	0.35	0.39	17
ASSAULT WITH DEADLY WEAPON ON POLICE OFFICER	1.00	0.67	0.80	9
ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT	0.81	0.85	0.83	365
ATTEMPTED ROBBERY	0.66	0.36	0.46	53
BATTERY - SIMPLE ASSAULT	0.89	0.92	0.90	622
BATTERY ON A FIREFIGHTER	0.50	0.50	0.50	2
BATTERY POLICE (SIMPLE)	0.84	0.80	0.82	20
BATTERY WITH SEXUAL CONTACT	0.58	0.76	0.66	46
BIKE - STOLEN	0.24	0.23	0.23	80
BOAT - STOLEN	0.00	1.00	0.00	0
BOMB SCARE	1.00	1.00	1.00	4
BRANDISH WEAPON	0.69	0.75	0.72	93
BUNCO, ATTEMPT	0.00	0.00	0.00	6
BUNCO, GRAND THEFT	0.74	0.63	0.68	94
BUNCO, PETTY THEFT	0.17	0.24	0.20	25
BURGLARY	0.71	0.68	0.70	391
BURGLARY FROM VEHICLE	0.79	0.76	0.77	578
CRIMES FROM VEHICLE, INDEMNIFICATION	0.26	0.20	0.23	2

THREATENING PHONE CALLS/LETTERS	0.00	0.00	0.00	7
THROWING OBJECT AT MOVING VEHICLE	0.29	0.22	0.25	9
TILL TAP - GRAND THEFT (\$950.01 OVER)	1.00	0.00	0.00	1
TILL TAP - PETTY (\$950 & UNDER)	1.00	0.00	0.00	1
TRESPASSING	0.55	0.61	0.57	119
UNAUTHORIZED COMPUTER ACCESS	0.17	0.33	0.22	3
VANDALISM - FELONY (\$400 & OVER, ALL CHURCH VANDALISMS)	0.64	0.63	0.63	431
VANDALISM - MISDEMEANOR (\$399 OR UNDER)	0.42	0.43	0.42	265
VEHICLE - ATTEMPT STOLEN	0.36	0.29	0.32	14
VEHICLE - STOLEN	0.79	0.88	0.83	522
VIOLATION OF COURT ORDER	0.26	0.22	0.24	64
VIOLATION OF RESTRAINING ORDER	0.50	0.58	0.54	84
VIOLATION OF TEMPORARY RESTRAINING ORDER	0.50	0.07	0.12	15
WEAPONS POSSESSION/BOMBING	0.00	0.00	0.00	1
accuracy			0.67	7277
macro avg	0.46	0.37	0.35	7277
weighted avg	0.66	0.67	0.66	7277

Αποτελέσματα:

- 67% total accuracy. Accuracy είναι το ποσοστό των περιπτώσεων που προβλέπονται σωστά από τον classifier.
- Weighted average of precision είναι 66%. Precision είναι ο λόγος των σωστών αναφερόμενων θετικών έναντι όλων των αναφερόμενων θετικών (στη περίπτωση μας ως θετικό θεωρείται ένας συγκεκριμένος τύπος εγκλήματος)
- Weighted average of recall είναι 67%. Recall είναι η αναλογία των σωστών αναφερόμενων θετικών έναντι όλων των πραγματικών θετικών.
- Weighted average of f1-score είναι 66%. Η βαθμολογία F1 είναι ο αρμονικός μέσος του precision και του recall.

Βλέπουμε ότι το δέντρο μας τα πάει αρκετά καλά. Μάλιστα, προβλέπει σωστά το 67% όλων των περιστατικών που έχουν οριστεί για testing (τα οποία είναι 7277). Θα δοκιμάσουμε τώρα κάποιες επεκτάσεις decision trees.

- Μπορούμε επίσης να οπτικοποιήσουμε το δέντρο, για να κατανοήσουμε πως λειτουργεί αφού δεν αποτελεί μαύρο κουτί. Το μεγάλο πλήθος των διαφορετικών κατηγορικών τιμών των στηλών του ωστόσο το κάνουν αρκετά δύσκολο. Εδώ παίρνουμε μια εικόνα της δομής ενός τμήματος του:

- Now we visualize the decision tree that was created
- It is visible that we can easily interpret the decision tree, thus they are also called white boxes

```
In [117]: temp = []
for item in X_test.columns:
    temp.append(item)

In [118]: text_representation = tree.export_text(clf_gini, feature_names= temp)
print(text_representation)

|--- unknown <= 0.50
|   |--- PART1_2_1 <= 0.50
|   |   |--- 0329 <= 0.50
|   |   |   |--- 2000 <= 0.50
|   |   |   |   |--- WEAPON_DESC_STRONG-ARM (HANDS, FIST, FEET OR BODILY FORCE) <= 0.50
|   |   |   |   |--- WEAPON_DESC_VERBAL THREAT <= 0.50
|   |   |   |   |--- 1822 <= 0.50
|   |   |   |   |   |--- WEAPON_DESC_unknown <= 0.50
|   |   |   |   |   |--- 1100 <= 0.50
|   |   |   |   |   |--- 1258 <= 0.50
|   |   |   |   |   |   |--- 0416 <= 0.50
|   |   |   |   |   |   |--- truncated branch of depth 20
|   |   |   |   |   |   |--- 0416 > 0.50
|   |   |   |   |   |   |--- truncated branch of depth 19
|   |   |   |   |   |--- 1258 > 0.50
|   |   |   |   |   |   |--- 0416 <= 0.50
|   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |--- 0416 > 0.50
|   |   |   |   |   |   |--- class: CHILD ABUSE (PHYSICAL) - SIMPLE ASSAULT
|   |   |   |   |   |--- 1100 > 0.50
|   |   |   |   |   |--- 1814 <= 0.50
|   |   |   |   |   |   |--- class: DISCHARGE FIREARMS/SHOTS FIRED
|   |   |   |   |   |--- 1814 > 0.50
|   |   |   |   |   |   |--- class: WEAPONS POSSESSION/BOMBING
|   |   |   |   |--- WEAPON_DESC_unknown > 0.50
```

AdaBoost classifier

Τώρα θα δοκιμάσουμε τον AdaBoost classifier. Boosting είναι μια άλλη βελτιωμένη μέθοδος classification. Συγκεκριμένα, εκπαιδεύουμε μια σειρά από αδύναμους learners όπως δέντρα αποφάσεων. Ο αλγόριθμος πηγαίνει:

- Εκπαιδεύουμε έναν ασθενή predictor, όπως ένα μικρό δέντρο αποφάσεων, στο σύνολο δεδομένων μας.
 - Λαμβάνουμε υπόψη τα σφάλματα στις προβλέψεις και αλλάζουμε τα βάρη στο εκπαιδευτικό μας σετ έτσι ώστε:
 - Τα βάρη των δεδομένων που είχαν προβλεφθεί σωστά μειώνονται.
 - Τα βάρη των δεδομένων που είχαν προβλεφθεί λάθος αυξάνονται
 - Επιστρέφουμε στο βήμα 1.
- Σημείωση: αυτά τα βήματα γίνονται μέσα στην python

AdaBoost classifier

Now AdaBoost classifier is used. Boosting is another method for improved classification. Particularly, we fit a sequence of weak learners, such as a very small decision trees. The algorithm goes:

1. We train a weak predictor, such as a small decision tree, on our dataset.
2. We take notice of the errors in the predictions and we reweigh our training set so that:
 - The weights of the data that were correctly predicted are decreased.
 - The weights of the data that were incorrectly predicted are increased.
3. We go back to step 1. Note: those steps are made inside pandas

```
In [83]: clf = AdaBoostClassifier(
    DecisionTreeClassifier(max_depth=None),
    n_estimators=100)
booster = clf.fit(X_train, np.ravel(y_train))
booster_predict = booster.predict(X_test)
```

- Τώρα μπορούμε να εξετάσουμε τα αποτελέσματα των προβλέψεων

AdaBoost results:

- 70% total accuracy.
- Weighted average of precision is 68%.
- Weighted average of recall is 70%.
- Weighted average of f1-score is 68%.

	precision	recall	f1-score	support
ARSON	0.57	0.47	0.52	17
ASSAULT WITH DEADLY WEAPON ON POLICE OFFICER	1.00	0.67	0.80	9
ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT	0.81	0.90	0.85	365
ATTEMPTED ROBBERY	0.67	0.34	0.45	53
BATTERY - SIMPLE ASSAULT	0.87	0.95	0.91	622
BATTERY ON A FIREFIGHTER	1.00	0.50	0.67	2
BATTERY POLICE (SIMPLE)	0.87	0.65	0.74	20
BATTERY WITH SEXUAL CONTACT	0.66	0.76	0.71	46
BIKE - STOLEN	0.36	0.26	0.30	80
BOAT - STOLEN	0.00	1.00	0.00	0
BOMB SCARE	0.80	1.00	0.89	4
BRANDISH WEAPON	0.72	0.76	0.74	93
BUNCO, ATTEMPT	1.00	0.00	0.00	6
BUNCO, GRAND THEFT	0.75	0.78	0.76	94
BUNCO, PETTY THEFT	0.24	0.24	0.24	25
BURGLARY	0.73	0.74	0.74	391
BURGLARY FROM VEHICLE	0.77	0.79	0.78	578
<hr/>				
THREATENING PHONE CALLS/LETTERS	1.00	0.14	0.25	7
THROWING OBJECT AT MOVING VEHICLE	1.00	0.22	0.36	9
TILL TAP - GRAND THEFT (\$950.01 & OVER)	1.00	0.00	0.00	1
TILL TAP - PETTY (\$950 & UNDER)	1.00	0.00	0.00	1
TRESPASSING	0.54	0.73	0.62	119
UNAUTHORIZED COMPUTER ACCESS	1.00	0.33	0.50	3
VANDALISM - FELONY (\$400 & OVER, ALL CHURCH VANDALISMS)	0.63	0.76	0.69	431
VANDALISM - MISDEMEANOR (\$399 OR UNDER)	0.44	0.29	0.35	265
VEHICLE - ATTEMPT STOLEN	0.60	0.21	0.32	14
VEHICLE - STOLEN	0.80	0.93	0.86	522
VIOLATION OF COURT ORDER	0.24	0.14	0.18	64
VIOLATION OF RESTRAINING ORDER	0.49	0.61	0.54	84
VIOLATION OF TEMPORARY RESTRAINING ORDER	0.33	0.07	0.11	15
WEAPONS POSSESSION/BOMBING	1.00	0.00	0.00	1
<hr/>				
accuracy			0.70	7277
macro avg	0.66	0.36	0.38	7277
weighted avg	0.68	0.70	0.68	7277

Αποτελέσματα:

- 70% total accuracy. Accuracy είναι το ποσοστό των περιπτώσεων που προβλέπονται σωστά από τον classifier.
- Weighted average of precision είναι 68%. Precision είναι ο λόγος των σωστών αναφερόμενων θετικών έναντι όλων των αναφερόμενων θετικών (στη περίπτωση μας ως θετικό θεωρείται ένας συγκεκριμένος τύπος εγκλήματος)
- Weighted average of recall είναι 70%. Recall είναι η αναλογία των σωστών αναφερόμενων θετικών έναντι όλων των πραγματικών θετικών.
- Weighted average of f1-score είναι 68%. Η βαθμολογία F1 είναι ο αρμονικός μέσος του precision και του recall.

Βλέπουμε ότι ο AdaBoost classifier τα πάει καλύτερα από το decision tree. Μάλιστα, προβλέπει σωστά το 70% όλων των περιστατικών που έχουν οριστεί για testing (τα οποία είναι 7277).

Bagging classifier

Τώρα θα χρησιμοποιήσουμε out of bag error estimation ή με άλλα λόγια των Bagging classifier. Η συγκεκριμένη μέθοδος χρησιμοποιεί το υπόλοιπο 1/3 των παρατηρήσεων που δεν χρησιμοποιούνται για να εκπαιδεύσουν το δέντρο απόφασης και αναφέρονται ως out of bag observations(OOB).

Μάλιστα, μπορούμε να προβλέψουμε ένα περιστατικό χρησιμοποιώντας τα δέντρα απόφασης στα οποία το συγκεκριμένο περιστατικό δεν χρησιμοποιήθηκε για εκπαίδευση. Ο τύπος εγκλήματος που προβλέπεται είναι αυτός με τους περισσότερους ψήφους (ο τύπος εγκλήματος που τα περισσότερα δέντρα προβλέπουν στα οποία δέντρα το περιστατικό ήταν out of bag observation)

Bagging classifier

Next up we use out-of-Bag Error Estimation. This estimation uses the remaining 1/3 of the observations that are not used to fit a given bagged tree and are referred to as the out-of-bag (OOB) observations.

- Like that, we can predict the response for the i th observation using each of the trees in which that observation was OOB.
- This will yield around $B/3$ predictions for the i th observation, where B is the number of bootstrapped training sets.
- To get a single prediction for the i th observation, we can take a majority vote (in classification trees).
- So we get a single prediction for the i th observation; we do the same for all n observations. In this way we can get an overall error estimate.

```
In [75]: bagging_crime_tree = BaggingClassifier(DecisionTreeClassifier(criterion='gini'),
                                             n_estimators=50,
                                             n_jobs=None)

bagging_crime_tree = bagging_crime_tree.fit(X_train, np.ravel(y_train))
baggin_predict = bagging_crime_tree.predict(X_test)
```

➤ Ας δούμε πως τα πήγε ο συγκεκριμένος classifier

Now we check how our decision tree performed

- 71 % total accuracy. Accuracy is the proportion of instances correctly classified by the classifier.
- Weighted average of precision is 69%. Precision is the ratio of correctly reported positives over all reported positives
- Weighted average of recall is 71%. Recall is the ratio of correctly reported positives over all actual positives
- Weighted average of f1-score is 68%. F1-score is the harmonic mean of the precision and the recall.

```
In [76]: print(classification_report(y_test, baggin_predict, zero_division=1))
```

	precision	recall	f1-score	support
ARSON	0.50	0.59	0.54	17
ASSAULT WITH DEADLY WEAPON ON POLICE OFFICER	1.00	0.67	0.80	9
ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT	0.80	0.90	0.85	365
ATTEMPTED ROBBERY	0.65	0.38	0.48	53
BATTERY - SIMPLE ASSAULT	0.87	0.95	0.91	622
BATTERY ON A FIREFIGHTER	1.00	0.50	0.67	2
BATTERY POLICE (SIMPLE)	0.76	0.80	0.78	20
BATTERY WITH SEXUAL CONTACT	0.62	0.76	0.69	46
BIKE - STOLEN	0.49	0.31	0.38	80
BOAT - STOLEN	0.00	1.00	0.00	0
BOMB SCARE	1.00	1.00	1.00	4
BRANDISH WEAPON	0.73	0.78	0.76	93
BUNCO, ATTEMPT	1.00	0.00	0.00	6
BUNCO, GRAND THEFT	0.73	0.85	0.78	94
BUNCO, PETTY THEFT	0.17	0.08	0.11	25
BURGLARY	0.74	0.73	0.74	391
BURGLARY FROM VEHICLE	0.75	0.77	0.76	578

THREATENING PHONE CALLS/LETTERS	0.00	0.00	0.00	7
THROWING OBJECT AT MOVING VEHICLE	0.50	0.11	0.18	9
TILL TAP - GRAND THEFT (\$950.01 & OVER)	1.00	0.00	0.00	1
TILL TAP - PETTY (\$950 & UNDER)	1.00	0.00	0.00	1
TRESPASSING	0.54	0.76	0.63	119
UNAUTHORIZED COMPUTER ACCESS	1.00	0.33	0.50	3
VANDALISM - FELONY (\$400 & OVER, ALL CHURCH VANDALISMS)	0.65	0.83	0.73	431
VANDALISM - MISDEAMEANOR (\$399 OR UNDER)	0.53	0.29	0.38	265
VEHICLE - ATTEMPT STOLEN	0.50	0.07	0.12	14
VEHICLE - STOLEN	0.80	0.97	0.87	522
VIOLATION OF COURT ORDER	0.30	0.12	0.18	64
VIOLATION OF RESTRAINING ORDER	0.49	0.67	0.57	84
VIOLATION OF TEMPORARY RESTRAINING ORDER	0.50	0.07	0.12	15
WEAPONS POSSESSION/BOMBING	1.00	0.00	0.00	1
accuracy			0.71	7277
macro avg	0.74	0.36	0.37	7277
weighted avg	0.69	0.71	0.68	7277

Αποτελέσματα:

- 71% total accuracy. Accuracy είναι το ποσοστό των περιπτώσεων που προβλέπονται σωστά από τον classifier.
- Weighted average of precision είναι 69%.
- Weighted average of recall είναι 71%.
- Weighted average of f1-score είναι 68%.

Βλέπουμε ότι ο Bagging classifier τα πάει ακόμα καλύτερα αφού προβλέπει σωστά το 71% όλων των περιστατικών που έχουν οριστεί για testing (τα οποία είναι 7277).

Random Forest

Τώρα θα χρησιμοποιήσουμε τα random forest classifier. Τα random forest είναι βελτιωμένη έκδοση των Bagging trees.

Random Forest

Next up random forest classifier will be used. Random forests are an improvement over bagged trees

- Random forests are used to decorrelate the trees
- Therefore, the forest will have uncorrelated quantities to achieve a reduction in variance

```
In [81]: forest = RandomForestClassifier(n_estimators=50, max_depth=None,
                                         min_samples_split=2)
forest = forest.fit(X_train, np.ravel(y_train)) #training data
forest_predict = forest.predict(X_test)
```

➤ Εδώ βλέπουμε τα αποτελέσματα του random forest

Random forest results:

- 70% total accuracy. Accuracy is the proportion of instances correctly classified by the classifier.
- Weighted average of precision is 67%. Precision is the ratio of correctly reported positives over all reported positives
- Weighted average of recall is 70%. Recall is the ratio of correctly reported positives over all actual positives
- Weighted average of f1-score is 65%. F1-score is the harmonic mean of the precision and the recall.

```
In [82]: print(classification_report(y_test, forest_predict, zero_division=1))
```

	precision	recall	f1-score	support
ARSON	0.50	0.06	0.11	17
ASSAULT WITH DEADLY WEAPON ON POLICE OFFICER	1.00	0.67	0.80	9
ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT	0.67	0.94	0.78	365
ATTEMPTED ROBBERY	0.50	0.02	0.04	53
BATTERY - SIMPLE ASSAULT	0.79	0.99	0.87	622
BATTERY ON A FIREFIGHTER	1.00	0.00	0.00	2
BATTERY POLICE (SIMPLE)	0.60	0.30	0.40	20
BATTERY WITH SEXUAL CONTACT	0.73	0.41	0.53	46
BIKE - STOLEN	0.52	0.21	0.30	80
BOAT - STOLEN	0.00	1.00	0.00	0
BOMB SCARE	0.67	0.50	0.57	4
BRANDISH WEAPON	0.85	0.56	0.68	93
BUNCO, ATTEMPT	1.00	0.00	0.00	6
BUNCO, GRAND THEFT	0.74	0.94	0.83	94
BUNCO, PETTY THEFT	0.50	0.08	0.14	25
BURGLARY	0.68	0.80	0.74	391
BURGLARY FROM VEHICLE	0.68	0.88	0.77	578

```
In [82]: print(classification_report(y_test, forest_predict, zero_division=1))
```

	accuracy	macro avg	weighted avg	
THREATENING PHONE CALLS/LETTERS	1.00	0.00	0.00	7
THROWING OBJECT AT MOVING VEHICLE	1.00	0.11	0.20	9
TILL TAF - GRAND THEFT (\$950.01 & OVER)	1.00	0.00	0.00	1
TILL TAF - PETTY (\$950 & UNDER)	1.00	0.00	0.00	1
TRESPASSING	0.60	0.72	0.66	119
UNAUTHORIZED COMPUTER ACCESS	1.00	0.00	0.00	3
VANDALISM - FELONY (\$400 & OVER, ALL CHURCH VANDALISMS)	0.61	0.88	0.72	431
VANDALISM - MISDEAMEANOR (\$399 OR UNDER)	0.51	0.18	0.26	265
VEHICLE - ATTEMPT STOLEN	1.00	0.00	0.00	14
VEHICLE - STOLEN	0.79	0.98	0.87	522
VIOLATION OF COURT ORDER	0.43	0.16	0.23	64
VIOLATION OF RESTRAINING ORDER	0.47	0.64	0.55	84
VIOLATION OF TEMPORARY RESTRAINING ORDER	0.67	0.13	0.22	15
WEAPONS POSSESSION/BOMBING	1.00	0.00	0.00	1
accuracy	0.70			
macro avg	0.79	0.29	0.30	7277
weighted avg	0.67	0.70	0.65	7277

Αποτελέσματα:

- 70% total accuracy. Accuracy είναι το ποσοστό των περιπτώσεων που προβλέπονται σωστά από τον classifier.
- Weighted average of precision είναι 67%.
- Weighted average of recall είναι 70%.
- Weighted average of f1-score είναι 65%.

Βλέπουμε ότι το random forest τα πάει ελαφρώς χειρότερα από τον Bagging classifier με 70% accuracy.

Extremely Randomized Trees

Τελευταίο classifier που θα δοκιμάσουμε είναι τα extremely randomized trees. Στο συγκεκριμένο classifier η τυχαιότητα πάει ένα βήμα παραπέρα.

Extremely Randomized Trees

Next up extremely randomized trees are used

- In extremely randomized trees, randomness goes one step further in the way splits are computed.
- As in random forests, a random subset of candidate features is used but thresholds are drawn at random for each candidate feature and the best of these randomly-generated thresholds and its corresponding feature is picked as the splitting rule.

```
In [79]: extremely_rts = ExtraTreesClassifier(n_estimators=100, criterion='gini',
                                         max_depth=None,
                                         min_samples_split=2)
extremely_rts = extremely_rts.fit(X_train, np.ravel(y_train)) #training
extreme_predict = extremely_rts.predict(X_test)
```

➤ Τώρα μπορούμε να δούμε τα αποτελέσματα

Extremely random forest results:

- 71% total accuracy. Accuracy is the proportion of instances correctly classified by the classifier.
- Weighted average of precision is 69%. Precision is the ratio of correctly reported positives over all reported positives
- Weighted average of recall is 71%. Recall is the ratio of correctly reported positives over all actual positives
- Weighted average of f1-score is 66%. F1-score is the harmonic mean of the precision and the recall.

```
In [80]: print(classification_report(y_test, extreme_predict, zero_division=1))
```

		precision	recall	f1-score	support
ARSON	0.67	0.12	0.20	17	
ASSAULT WITH DEADLY WEAPON ON POLICE OFFICER	1.00	0.67	0.80	9	
ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT	0.73	0.93	0.82	365	
ATTEMPTED ROBBERY	1.00	0.00	0.00	53	
BATTERY - SIMPLE ASSAULT	0.80	0.99	0.89	622	
BATTERY ON A FIREFIGHTER	1.00	0.00	0.00	2	
BATTERY POLICE (SIMPLE)	0.69	0.45	0.55	20	
BATTERY WITH SEXUAL CONTACT	0.66	0.67	0.67	46	
BIKE - STOLEN	0.59	0.29	0.39	80	
BOAT - STOLEN	0.00	1.00	0.00	0	
BOMB SCARE	1.00	0.50	0.67	4	
BRANDISH WEAPON	0.85	0.54	0.66	93	
BUNCO, ATTEMPT	1.00	0.00	0.00	6	
BUNCO, GRAND THEFT	0.72	0.90	0.80	94	
BUNCO, PETTY THEFT	0.33	0.08	0.13	25	
BURGLARY	0.69	0.82	0.75	391	
BURGLARY FROM VEHICLE	0.69	0.88	0.77	578	
BURGLARY FROM VEHICLE, INSTRUMENT	1.00	0.00	0.00	6	

THREATENING PHONE CALLS/LETTERS	1.00	0.14	0.25	7	--
THROWING OBJECT AT MOVING VEHICLE	1.00	0.11	0.20	9	
TILL TAP - GRAND THEFT (\$950.01 & OVER)	1.00	0.00	0.00	1	
TILL TAP - PETTY (\$950 & UNDER)	1.00	0.00	0.00	1	
TRESPASSING	0.60	0.76	0.67	119	
UNAUTHORIZED COMPUTER ACCESS	1.00	0.33	0.50	3	
VANDALISM - FELONY (\$400 & OVER, ALL CHURCH VANDALISMS)	0.62	0.88	0.73	431	
VANDALISM - MISDEAMEANOR (\$399 OR UNDER)	0.53	0.19	0.28	265	
VEHICLE - ATTEMPT STOLEN	1.00	0.00	0.00	14	
VEHICLE - STOLEN	0.79	0.97	0.87	522	
VIOLATION OF COURT ORDER	0.41	0.11	0.17	64	
VIOLATION OF RESTRAINING ORDER	0.47	0.75	0.58	84	
VIOLATION OF TEMPORARY RESTRAINING ORDER	1.00	0.13	0.24	15	
WEAPONS POSSESSION/BOMBING	1.00	0.00	0.00	1	
accuracy			0.71	7277	
macro avg	0.82	0.31	0.34	7277	
weighted avg	0.69	0.71	0.66	7277	

Αποτελέσματα:

- 71% total accuracy.
- Weighted average of precision είναι 69%.
- Weighted average of recall είναι 71%.
- Weighted average of f1-score είναι 66%.

Βλέπουμε ότι τα extremely randomized trees τα πάνε ελαφρώς καλύτερα από το random forest αφού πετυχαίνουν 71% accuracy.

Συμπεράσματα

Με βάση τις μετρήσεις accuracy, οι καλύτεροι classifiers είναι τα extremely randomized trees και ο Bagging classifier, και οι δύο με 71%. Αυτό σημαίνει ότι και οι δύο θα προβλέψουν με ακρίβεια περίπου 7 στα 10 περιστατικά όπου λείπει η περιγραφή του εγκλήματος. Αναλυτικά, και οι δύο αυτοί classifiers έχουν τον ίδιο weighted average of precision και recall, αλλά ο Bagging classifier έχει ελαφρώς υψηλότερο weighted average of f1-score.

Πολύ σημαντικό συνιστά ότι πέραν του accuracy και γνωρίζοντας τους ορισμούς precision και recall, μπορούμε να χρησιμοποιήσουμε τις τελευταίες αυτές μετρικές να βρούμε τον καλύτερο classifier για συγκεκριμένες περιπτώσεις. Έστω για παράδειγμα ότι θέλουμε να χρησιμοποιήσουμε τον classifier που για το συγκεκριμένο έγκλημα 'ROBBERY' δεν χάνει κανένα περιστατικό. Με άλλα λόγια, θέλουμε τον classifier που σε κάθε περιστατικό robbery που καλείται να προβλέψει, προβλέπει σωστά ότι το συγκεκριμένο περιστατικό είναι robbery. Για το συγκεκριμένο παράδειγμα πρέπει απλά να βρούμε το υψηλότερο σκορ recall για τον τύπο εγκλήματος robbery μεταξύ όλων των classifiers που έχουμε δημιουργήσει. Αυτό συμβαίνει αφού το recall στη συγκεκριμένη περίπτωση ορίζεται ως η αναλογία των σωστά αναφερόμενων robberies έναντι όλων των πραγματικών robberies. Αφού ελέγχουμε τους πίνακες με τα αποτελέσματα, ο classifier των extremely randomized trees προσφέρει το υψηλότερο recall για τον τύπο

εγκλήματος robbery με 88%

RESISTING ARREST	0.55	0.35	0.43	17
ROBBERY	0.74	0.88	0.80	281
SEX OFFENDER REGISTRANT OUT OF COMPLIANCE	0.57	0.57	0.57	7

Έστω τώρα θέλουμε να βρούμε τώρα τον classifier που για το συγκεκριμένο έγκλημα 'RAPE, FORCIBLE' δεν το προβλέπει λάθος. Με άλλα λόγια, ψάχνουμε τον classifier που σε κάθε πρόβλεψη 'rape, forcible' που κάνει, πέφτει έξω τις λιγότερες φορές, δηλαδή τα περιστατικά που προβλέπει ως rape, forcible να είναι όντως 'rape, forcible'. Για το συγκεκριμένο παράδειγμα πρέπει απλά να βρούμε το υψηλότερο σκορ precision για τον τύπο εγκλήματος rape,forcible μεταξύ όλων των classifiers που έχουμε δημιουργήσει. Αυτό συμβαίνει αφού το precision στη συγκεκριμένη περίπτωση ορίζεται ως ο λόγος των σωστών αναφερόμενων rape,forcible έναντι όλων των αναφερόμενων rape,forcible. Μετά από έλεγχο των πινάκων με τα αποτελέσματα, ο classifier Bagging έχει το υψηλότερο precision για τον τύπο εγκλήματος rape,forcible με 93%.

RAPE, ATTEMPTED	1.00	0.50	0.67	2
RAPE, FORCIBLE	0.93	0.82	0.87	34
RECKLESS DRIVING	1.00	0.00	0.00	1

Πιθανότητες

Μια πολύ καλή χρήση των decision trees είναι ότι μπορούν να υπολογίσουν την πιθανότητα μιας παρατήρησης (περιστατικού) να ανήκει σε οποιαδήποτε class (τύπος εγκλήματος). Στην πραγματικότητα, ο τύπος εγκλήματος που προβλέπεται για ένα περιστατικό είναι αυτός με την υψηλότερη πιθανότητα από όλους τους υπόλοιπους.

Για τη συγκεκριμένη ανάλυση, επιλέγουμε τα extremely randomized trees που αποδίδει ένα από τα καλύτερα accuracies.

Πρώτο περιστατικό

- Ας δούμε τις πιθανότητες του πρώτου περιστατικού του testing data. Η λίστα περιέχει τόσους αριθμούς όσοι είναι και οι τύποι εγκλημάτων που μπορεί να προβλέψει (δηλαδή όλους τους διαφορετικούς τύπους εγκλημάτων που συνάντησε στα training data). Κάθε αριθμός μέσα στη πρώτη λίστα είναι μια πιθανότητα και αντιστοιχεί στον τύπο εγκλήματος που βρίσκεται στην ίδια θέση της δεύτερης λίστας.

Σημείωση: Ο πίνακας των πιθανοτήτων βλέπουμε ότι αθροίζει πάντα στην μονάδα. Επίσης, η λίστα των εγκλημάτων έχει κοπεί αφού δεν χωράει στην οθόνη.

```
First incident
Let's see the probabilities for the first incident of the test data. Each list's item is a probability and the index of the item corresponds to the crime type that is found in extremely_rts.classes_
In [119]: index= 1
X = X_test[index-1 : index]
position = np.where(extremely_rts.predict_proba(X) [0] == max(extremely_rts.predict_proba(X) [0]))
print(extremely_rts.predict_proba(X) [0])
extremely_rts.classes_
[0.02 0.   0.5 0.   0.03 0.   0.   0.   0.   0.   0.   0.05 0.
 0.   0.   0.02 0.01 0.   0.03 0.   0.   0.   0.   0.   0.   0.   0.
 0.   0.   0.   0.   0.01 0.03 0.   0.   0.   0.   0.   0.   0.   0.
 0.   0.   0.   0.   0.   0.   0.   0.01 0.   0.   0.   0.   0.   0.
 0.   0.   0.   0.   0.02 0.   0.   0.   0.01 0.   0.   0.   0.01 0.
 0.01 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.04 0.   0.
 0.02 0.   0.   0.   0.03 0.01 0.   0.   0.   0.   0.   0.   0.   0.01
 0.   0.01 0.   0.05 0.   0.   0.04 0.   0.   0.   0.   0.   0.   0.
 0.   0.01 0.   0.   0.01 0.01 0.   0.   ]
```

```
Out[119]: array(['ARSON', 'ASSAULT WITH DEADLY WEAPON ON POLICE OFFICER',
       'ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT',
       'ATTEMPTED ROBBERY', 'BATTERY - SIMPLE ASSAULT',
       'BATTERY ON A FIREFIGHTER', 'BATTERY POLICE (SIMPLE)',
       'BATTERY WITH SEXUAL CONTACT', 'BIGAMY', 'BIKE - STOLEN',
       'BOAT - STOLEN', 'BOMB SCARE', 'BRANDISH WEAPON', 'BUNCO, ATTEMPT',
       'BUNCO, GRANT THEFT', 'BUNCO, PETTY THEFT', 'BURGLARY',
       'BURGLARY FROM VEHICLE', 'BURGLARY FROM VEHICLE, ATTEMPTED',
       'BURGLARY, ATTEMPTED', 'CHILD ABANDONMENT',
       'CHILD ABUSE (PHYSICAL) - AGGRAVATED ASSAULT',
       'CHILD ABUSE (PHYSICAL) - SIMPLE ASSAULT',
       'CHILD ANNOYING (17YRS & UNDER)', 'CHILD NEGLECT (SEE 300 W.I.C.)',
       'CHILD PORNOGRAPHY', 'CHILD STEALING', 'CONSPIRACY',
       'CONTEMPT OF COURT', 'CONTRIBUTING', 'COUNTERFEIT',
       'CREDIT CARDS, FRAUD USE ($950 & UNDER',
       'CREDIT CARDS, FRAUD USE ($950.01 & OVER)', 'CRIMINAL HOMICIDE',
       'CRIMINAL THREATS - NO WEAPON DISPLAYED',
       'CRM AGNST CHLD (13 OR UNDER) (14-15 & SUSP 10 YRS OLDER)',
       'CRUELTY TO ANIMALS',
       'DEFRAUDING INNKEEPER/THEFT OF SERVICES, $400 & UNDER',
       'DEFRAUDING INNKEEPER/THEFT OF SERVICES, OVER $400',
       'DISCHARGE FIREARMS/SHOTS FIRED',
       'DISHONEST EMPLOYEE - GRAND THEFT',
       'DISHONEST EMPLOYEE - PETTY THEFT',
       'DISHONEST EMPLOYEE ATTEMPTED THEFT', 'DISTURBING THE PEACE',
       'DOCUMENT FORGERY / STOLEN FELONY',
       'DOCUMENT WORTHLESS ($200 & UNDER')]
```

- Τώρα μπορούμε να δούμε τον τύπο του εγκλήματος που προβλέφθηκε και τις πιθανότητες που του δίνει ο classifier. Συγκεκριμένα, προβλέφθηκε το περιστατικό να είναι 'ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT' με πιθανότητα 50%

- Here we can see the crime type that is predicted (the crime type that corresponds to the highest probability) together with the probability

```
In [125]: print(extremely_rts.classes_[position])
extremely_rts.predict_proba(X)[0][position]
['ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT']

Out[125]: array([0.5])
```

- Τώρα μπορούμε να δούμε τι έγκλημα ήταν στην πραγματικότητα το περιστατικό. Βλέπουμε ότι όντως η πρόβλεψη ήταν σωστή.

- And this is the correct crime type
- We see that the prediction is correct

```
In [126]: y_test.head(1)

Out[126]: 26126    ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT
Name: CRM_DESC, dtype: object
```

Δεύτερο περιστατικό

- Ας εξετάσουμε τώρα το δεύτερο περιστατικό. Βλέπουμε των πίνακα των πιθανοτήτων εδώ:

Second incident

Let's see the probabilities for the second incident of the test data

```
In [127]: index= 2
X = X_test[index-1 : index]
position = np.where(extremely_rts.predict_proba(X)[0] == max(extremely_rts.predict_proba(X)[0]))
print(extremely_rts.predict_proba(X)[0])

[0.  0.   0.01 0.  0.   0.   0.   0.   0.42 0.  0.   0.   0.
 0.  0.   0.16 0.06 0.  0.   0.   0.   0.   0.   0.   0.   0.
 0.  0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
 0.  0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
 0.  0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
 0.  0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
 0.  0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
 0.  0.   0.   0.   0.   0.   0.01 0.  0.   0.   0.   0.   0.01 0.05
 0.  0.   0.   0.19 0.  0.   0.08 0.  0.   0.   0.   0.   0.   0.
 0.  0.01 0.  0.   0.   0.   0.   0.   0.   0.   0.   0.   0. ]
```

- Ο classifier προβλέπει ότι το περιστατικό είναι 'BIKE-STOLEN' με πιθανότητα 42%

- The prediction is a stolen bike with a probability of 42%

```
In [128]: print(extremely_rts.classes_[position])
extremely_rts.predict_proba(X)[0][position]

['BIKE - STOLEN']

Out[128]: array([0.42])
```

- Πάλι βλέπουμε ότι η πρόβλεψη είναι σωστή

- The actual crime committed is seen here
- The prediction is once again accurate

```
In [130]: y_test[index-1 : index]

Out[130]: 29416    BIKE - STOLEN
Name: CRM_DESC, dtype: object
```

Τρίτο Περιστατικό

- Τώρα θα εξετάσουμε το τρίτο περιστατικό των testing data. Εδώ φαίνεται ο πίνακας των πιθανοτήτων

Third incident

Now we will check for the third incident of the test data. We can see the probability table

```
In [131]: index= 3
X = X_test[index-1 : index]
position = np.where(extremely_rts.predict_proba(X) [0] == max(extremely_rts.predict_proba(X) [0]))
print(extremely_rts.predict_proba(X) [0])

[0.   0.   0.01 0.  0.01 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
 0.   0.   0.03 0.01 0.  0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
 0.01 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
 0.   0.   0.02 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
 0.01 0.   0.01 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
 0.01 0.   0.01 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
 0.   0.   0.01 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
 0.61 0.22 0.   0.   0.02 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.02 0.]
```

- Ο classifier προβλέπει ότι το περιστατικό είναι 'VANDALISM - FELONY (\$400 & OVER, ALL CHURCH VANDALISMS)' με πιθανότητα 61%

- The incident is predicted to be a vandalism - felony of 400 and more dollars with the probability of 61 %

```
In [132]: print(extremely_rts.classes_[position])
extremely_rts.predict_proba(X) [0][position]

['VANDALISM - FELONY ($400 & OVER, ALL CHURCH VANDALISMS)']

Out[132]: array([0.61])
```

- Αυτή τη φορά ο classifier πρόβλεψε λάθος αφού το περιστατικό είναι στην πραγματικότητα 'VANDALISM - MISDEAMEANOR (\$399 OR UNDER)'

- The incident is actually vandalism - misdeameanor of 399 and less dollars

```
In [110]: y_test[index-1 : index]
Out[110]: 20309    VANDALISM - MISDEAMEANOR ($399 OR UNDER)
Name: CRM_DESC, dtype: object
```

- Τώρα εξετάζουμε την δεύτερη μεγαλύτερη πιθανότητα στον πίνακα πιθανοτήτων του περιστατικού. Αυτή είναι 22% και μάλιστα αντιστοιχεί στον τύπο εγκλήματος 'VANDALISM - MISDEAMEANOR (\$399 OR UNDER)' το οποίο είναι και το έγκλημα που διαπράχθηκε στην πραγματικότητα.

- Let's find the second highest probability

```
In [116]: position2 = np.where(extremely_rts.predict_proba(X) [0] == sorted(extremely_rts.predict_proba(X) [0]) [-2])
sorted(extremely_rts.predict_proba(X) [0]) [-2]

Out[116]: 0.22
```

- We see that the second highest probability leads to the actual crime (vandalism - misdeameanor of 399 and less dollars)

```
In [118]: extremely_rts.classes_[position2]
Out[118]: array(['VANDALISM - MISDEAMEANOR ($399 OR UNDER)'), dtype=object)
```

Συμπέρασμα

Αυτό σημαίνει ότι παρόλο που το extremely randomized trees δεν προέβλεψε την σωστή τιμή στην αρχή, μας έδωσε μια ένδειξη ότι το περιστατικό θα μπορούσε επίσης να είναι vandalism - misdeamenor. Σε αυτό το συγκεκριμένο παράδειγμα είναι πολύ δύσκολο να διακρίνουμε το πραγματικό έγκλημα από τα άλλα δεδομένα που έχουμε στην διάθεση μας. Η πράξη του εγκλήματος είναι στην πραγματικότητα η ίδια, αλλά το ποσό της ζημιάς που προκλήθηκε ήταν ο διαφοροποιητής.

Μάλιστα, εάν δινόταν μια δεύτερη ευκαιρία στον classifier να προβλέψει, θα προέβλεπε με ακρίβεια. Αυτό είναι πολύ σημαντικό γιατί παρόλο που ο classifier δεν προβλέπει πάντα σωστά, μπορεί να επισημάνει άλλες πιθανές εκβάσεις. Επιπρόσθετα, ακόμη και για τους ανθρώπους, θα ήταν εξαιρετικά δύσκολο να διακρίνουμε τον ακριβή τύπο εγκλήματος που διαπράχθηκε, μόνο με την ανάλυση των υπόλοιπων πληροφοριών από τα περιστατικά. Εν κατακλείδι, μπορούμε να παρατηρήσουμε τη δύναμη της μηχανικής μάθησης με αυτήν την εφαρμογή ανάλυσης δεδομένων και προβλέψεων και συγκεκριμένα των decision trees και των επεκτάσεων τους.

ΕΠΙΛΟΓΟΣ

Με την συγκεκριμένη ανάλυση γίνεται κατανοητή η δύναμη και η σημασία της επιχειρηματικής ευφυΐας. Από ένα απλό dataset, μετά από πολλές επεξεργασίες, διαδικασίες και αλγορίθμους καταφέραμε να βγάλουμε πολύτιμη γνώση και ευφυΐα. Με τις πιλούσιες οπτικοποιήσεις μπορεί κανείς να κατανοήσει και να μάθει πολλά για την εγκληματικότητα του Los Angeles. Παράλληλα, με τις λειτουργίες εξόρυξης δεδομένων γίνεται κατανοητή η αποτελεσματικότητα και οι αμέτρητες δυνατότητες που προσφέρει η μηχανική μάθηση. Όπως ο CEO Jim Bergeson ανέφερε το 2016 «Τα δεδομένα θα μιλήσουν αν είσαι πρόθυμος να τα ακούσεις».

Βιβλιογραφία

Cambridge University Press. (2008). *K-means*. <https://nlp.stanford.edu/IR-book/html/htmledition/k-means-1.html>

Crime Data from 2010 to 2019 (official), <https://data.lacity.org/Public-Safety/Crime-Data-from-2010-to-2019/63jg-8b9z>

Dunham, M. H. (2006). *Data mining: Introductory and advanced topics*. Pearson Education India.

Stone Ward Staff, *Jim Bergeson quote*, <https://www.stoneward.com/blog/2016/04/glance-management-reporting/>

Λουρίδας, 2020, *Decision Trees and Ensemble Models*, lecture notes – [notebook](#), Applied Machine learning prediction model, Athens University of Economics and Business, delivered December 2020