Zewail City of Science, Technology, and Innovation
University of Science and Technology

Communications and Information

Engineering

Technical Documentation

Statistical Inference and Data Analysis - FALL2022

Prepared By

Abdullah Kamal Muhammad 201801271
Ibrahim Salah Abdelaal 201800224
Mohamed Abdelwahed 201801978
Mohamed Nasr 201801675

Supervised By
"Dr. Mahmoud Abdelaziz"
Eng Asmaa
Eng Anhar

2022/2023

Data Collection:

There were 4 online resources to access and download the data

- The national crime victimization survey (NCVS) data:
- NIBRS Reported offense count data:
- Recidivism data for the state of Georgia [2013-2015]
- Firearm laws per state

About these Dataset:

1- **The national crime victimization survey (NCVS) data:**

- **General Information & Downloading:**

- The National Crime Victimization Survey (NCVS) SODA API is a RESTful web service that provides data on violent and property victimization by selecting victim, household, and incident characteristics.
- All calls to the NCVS API are formed by adding the resource path (resource identifier) to the following base URL: https://data.ojp.usdoj.gov/resource/.

The NCVS API can return data to users in multiple formats including JSON, XML and CSV formats. To specify a desired format, append a format string (.json, for example) to the end of the URL; JSON is the default format:

- https://data.ojp.usdoj.gov/resource/{*dataset identifier*}.json
- https://data.ojp.usdoj.gov/resource/{*dataset identifier*}.csv

2- **<u>NIBRS Reported offense count data:</u>**

- **General Information:**
    - Source: National Incident-Based Reporting System (NIBRS) and FBI Crime Data API
    - Advantages: NIBRS goes much deeper because of its ability to provide circumstances and context for crimes like location, time of day, and whether the incident was cleared.
    - The FBI has made nationwide implementation of NIBRS a top priority
    - The FBI Crime Data API is a read-only web service that returns JSON or CSV data
    - SRS data is the legacy format that provides aggregated counts of the reported crime offenses known to law enforcement by location
    - Using API to access data: The API was designed to provide as much information as possible in a usable format

- **Dataset size:**
    - 835 rows
    - 4 columns
- **Downloading:**
    - By using the NCVS data API and reading a FBI JASON

3- **Recidivism data for the state of Georgia:**

- **General Information:**

- Data Provided by Georgia Department of Community Supervision, Georgia Crime Information Center
- Date of Data Creation: July 15, 2021
- Last Update: June 16, 2021
- Publisher: UDOJ / OJP / NIJ
- Contact Name: Joel Hunt
- Access Rights Category: Public
- Geographic Coverage Description: State of Georgia, Combination of PUMAs
- Category: Courts

- **Dataset size:**
    - 25.8K Rows
    - 54 Columns
- Each row represents a **Person**
- **Downloading:**
    - By using the NCVS data API (Available without an API key), to download as much of the Personal crime victimization and Personal population data as is available for all years (with limit >= 1000000).

- **URLs & API & JASON files**:
- api_url='https://data.ojp.usdoj.gov/resource/gcuy-rt5g.csv?$limit=1500000'
- http://content/FBI_Crime.json
- Url: 'https://api.usa.gov/crime/fbi/sapi/api/data/nibrs/{}/offender/states/{}/COUNT?API_KEY=rls0AqcJ4ZYcF8w1RIi0kBtcWYg791OohyiG4CgK'
- state_abr_name = requests.get("https://api.usa.gov/crime/fbi/sapi/api/agencies?API_KEY=rls0AqcJ4ZYcF8w1RIi0kBtcWYg791OohyiG4CgK")

Data cleaning:

- **NCVS:**
    - This data set needs to be cleaner due to the level of encryption it was written by
    - There some columns with ununderstandable names so they need to be renamed
    - Some preprocessing on the dataset was performed to check the possibility of needing cleaning data steps

Examples of ununderstandable columns names:

```
df_NCVS.describe()
```

ager	sex	hispanic	race	race_ethnicity	hincome1	hincome2	...	weapcat
63465.000000	63465.000000	63465.000000	63465.000000	63465.000000	63465.000000	63465.000000	...	63465.000000
3.067470	1.483164	2.415835	1.304703	1.907981	13.849980	-0.443945	...	0.945891
1.426953	0.499720	6.824125	0.785357	1.718922	26.714528	1.343024	...	1.624882
1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	-1.000000	...	0.000000
2.000000	1.000000	2.000000	1.000000	1.000000	3.000000	-1.000000	...	0.000000
3.000000	1.000000	2.000000	1.000000	1.000000	5.000000	-1.000000	...	0.000000
4.000000	2.000000	2.000000	1.000000	2.000000	7.000000	-1.000000	...	1.000000
6.000000	2.000000	88.000000	5.000000	6.000000	88.000000	5.000000	...	5.000000

The total number of columns is 38 with ununderstandable names and unimportant columns that can be deleted or dropped

Deleted Columns:

‘Idper','yearq','race_ethnicity','newcrime','region','citizen','msa','race','hincome2','newoff','seriousviolent','notify','vicservices','marital','locality','weapcat','injury','serious','treatment','locality','locality','educatn2','veteran','locationr','offtracenew','newwgt'

After deleting them:

The total number of columns changed to be only 14 columns:

	Unnamed: 0	year	ager	sex	hispanic	hincome1	popsize	educatn1	direl	weapon	offenderage	offendersex	wgtviccy	series
0	0	2004	2	2	2	1	1	4	1	2	3	1	1952.973730	1
1	1	2009	1	1	2	7	1	4	4	1	5	1	5570.687730	1
2	2	2004	4	1	2	5	0	4	3	2	4	1	3366.957480	1
3	3	2011	3	1	1	5	3	4	5	2	88	1	6991.560610	1
4	4	2004	2	1	2	6	1	5	4	2	3	3	2834.649050	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
63460	63460	2021	2	2	2	7	0	5	4	2	4	1	1255.609375	1
63461	63461	2021	4	2	2	7	1	5	4	2	3	1	842.529114	1
63462	63462	2021	1	1	2	7	1	3	3	2	1	1	1029.867432	1
63463	63463	2021	2	1	2	6	1	4	4	1	2	1	5833.862305	1
63464	63464	2021	4	1	1	7	1	5	4	1	5	3	2835.449463	1

63465 rows × 14 columns

- Renaming columns and replace the misleading names

	Unnamed: 0	year	ager	sex	hispanic	hincome1	popsize	educatn1	direl	weapon	offenderage	offendersex	wgtviccy	series
0	0	2004	18:24	Female	Non-hispanic	Less than $7,500	<100,000	High school	Intimates	No	18-29	Male	1952.973730	Not a series crime
1	1	2009	12:17	Male	Non-hispanic	$75,000 or more	<100,000	High school	Strangers	Yes	various ages offenders	Male	5570.687730	Not a series crime
2	2	2004	35:49	Male	Non-hispanic	$35,000 to $49,999	Not a place	High school	Well known/casual acquaintance	No	>=30	Male	3366.957480	Not a series crime
3	3	2011	25:34	Male	Hispanic	$35,000 to $49,999	250,000-499,999	High school	Do not know relationship	No	Residue	Male	6991.560610	Not a series crime
4	4	2004	18:24	Male	Non-hispanic	$50,000 to $74,999	<100,000	College	Strangers	No	18-29	Both	2834.649050	Not a series crime
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
63460	63460	2021	18:24	Female	Non-hispanic	$75,000 or more	Not a place	College	Strangers	No	>=30	Male	1255.609375	Not a series crime
63461	63461	2021	35:49	Female	Non-hispanic	$75,000 or more	<100,000	College	Strangers	No	18-29	Male	842.529114	Not a series crime
63462	63462	2021	12:17	Male	Non-hispanic	$75,000 or more	<100,000	Middle school	Well known/casual acquaintance	No	=<11	Male	1029.867432	Not a series crime
63463	63463	2021	18:24	Male	Non-hispanic	$50,000 to $74,999	<100,000	High school	Strangers	Yes	12:17	Male	5833.862305	Not a series crime
63464	63464	2021	35:49	Male	Hispanic	$75,000 or more	<100,000	College	Strangers	Yes	various ages offenders	Both	2835.449463	Not a series crime

```
NCVS['ager'] = NCVS['ager'].replace([1,
NCVS['sex'] = NCVS['sex'].replace([1, 2
NCVS['hispanic'] = NCVS['hispanic'].rep
NCVS['hincome1'] = NCVS['hincome1'].rep
NCVS['popsize'] = NCVS['popsize'].repla
NCVS['direl'] = NCVS['direl'].replace([
NCVS['educatn1']=NCVS['educatn1'].repla
NCVS['weapon']=NCVS['weapon'].replace([
NCVS['offenderage']=NCVS['offenderage']
NCVS['offendersex']=NCVS['offendersex']
NCVS['series']=NCVS['series'].replace([
NCVS
```

After the renaming process:

	Unnamed: 0	year	ager	sex	hispanic	Annual household income	popsize	Education level	Victim-offender relationship	weapon	offenderage	offendersex	Victimization weight	series
0	0	2004	18:24	Female	Non-hispanic	Less than $7,500	<100,000	High school	Intimates	No	18-29	Male	1952.973730	Not a series crime
1	1	2009	12:17	Male	Non-hispanic	$75,000 or more	<100,000	High school	Strangers	Yes	various ages offenders	Male	5570.687730	Not a series crime
2	2	2004	35:49	Male	Non-hispanic	$35,000 to $49,999	Not a place	High school	Well known/casual acquaintance	No	>=30	Male	3366.957480	Not a series crime
3	3	2011	25:34	Male	Hispanic	$35,000 to $49,999	250,000-499,999	High school	Do not know relationship	No	Residue	Male	6991.560610	Not a series crime
4	4	2004	18:24	Male	Non-hispanic	$50,000 to $74,999	<100,000	College	Strangers	No	18-29	Both	2834.649050	Not a series crime
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
63460	63460	2021	18:24	Female	Non-hispanic	$75,000 or more	Not a place	College	Strangers	No	>=30	Male	1255.609375	Not a series crime
63461	63461	2021	35:49	Female	Non-hispanic	$75,000 or more	<100,000	College	Strangers	No	18-29	Male	842.529114	Not a series crime
63462	63462	2021	12:17	Male	Non-hispanic	$75,000 or more	<100,000	Middle school	Well known/casual acquaintance	No	=<11	Male	1029.867432	Not a series crime
63463	63463	2021	18:24	Male	Non-hispanic	$50,000 to $74,999	<100,000	High school	Strangers	Yes	12:17	Male	5833.862305	Not a series crime
63464	63464	2021	35:49	Male	Hispanic	$75,000 or more	<100,000	College	Strangers	Yes	various ages offenders	Both	2835.449463	Not a series crime

63465 rows x 14 columns

- ## Recidivism Data:

It is a big dataset that consists of 25835 rows × 55 columns.

	Unnamed: 0	id	gender	race	age_at_release	residence_puma	gang_affiliated	supervision_risk_score_first	supervision_level_first	education_level	...	drugtests_meth_positive	dru
0	0	1	M	BLACK	43-47	16	False	3.0	Standard	At least some college	...	0.000000	
1	1	2	M	BLACK	33-37	16	False	6.0	Specialized	Less than HS diploma	...	0.000000	
2	2	3	M	BLACK	48 or older	24	False	7.0	High	At least some college	...	0.166667	
3	3	4	M	WHITE	38-42	16	False	7.0	High	Less than HS diploma	...	0.000000	
4	4	5	M	WHITE	33-37	16	False	4.0	Specialized	Less than HS diploma	...	0.058824	
...	...	...	...	...	...	...	...	...	...	...	...	...	
25830	25830	26756	M	BLACK	23-27	9	False	5.0	Standard	At least some college	...	0.000000	
25831	25831	26758	M	WHITE	38-42	25	False	5.0	Standard	At least some college	...	0.000000	
25832	25832	26759	M	BLACK	33-37	15	False	5.0	Standard	At least some college	...	NaN	
25833	25833	26760	F	WHITE	33-37	15	NaN	5.0	Standard	At least some college	...	0.000000	
25834	25834	26761	M	WHITE	28-32	12	False	5.0	Standard	High School Diploma	...	0.000000	

25835 rows × 55 columns

- **Checking columns names:**

This process did not detect any ununderstandable column name, so there is no need to rename or replace any name:

Columns:

'id', 'gender', 'race', 'age_at_release', 'residence_puma', 'gang_affiliated', 'supervision_risk_score_first', 'supervision_level_first', 'education_level', 'dependents', 'prison_offense', 'prison_years', 'prior_arrest_episodes_felony', 'prior_arrest_episodes_misd', 'prior_arrest_episodes_violent', 'prior_arrest_episodes_property', 'prior_arrest_episodes_drug', 'prior_arrest_episodes', 'prior_arrest_episodes_1', 'prior_arrest_episodes_2', 'prior_conviction_episodes', 'prior_conviction_episodes_1', 'prior_conviction_episodes_2', 'prior_conviction_episodes_3', 'prior_conviction_episodes_4', 'prior_conviction_episodes_5', 'prior_conviction_episodes_6', 'prior_conviction_episodes_7', 'prior_revocations_parole', 'prior_revocations_probation', 'condition_mh_sa', 'condition_cog_ed', 'condition_other', 'violations', 'violations_instruction', 'violations_failtoreport', 'violations_1', 'delinquency_reports', 'program_attendances', 'program_unexcusedabsences', 'residence_changes', 'avg_days_per_drugtest', 'drugtests_thc_positive', 'drugtests_cocaine_positive', 'drugtests_meth_positive', 'drugtests_other_positive', 'percent_days_employed', 'jobs_per_year', 'employment_exempt', 'recidivism_within_3years', 'recidivism_arrest_year1', 'recidivism_arrest_year2', 'recidivism_arrest_year3', 'training_sample'

Firearm Data:

This process did not detect any ununderstandable column name, so there is no need to rename or replace any name:

Columns:

	state	year	felony	invcommitment	invoutpatient	danger	drugmisdemeanor	alctreatment	alcoholism	relinquishment	...	expartedating
0	Alabama	1991	0	0	0	0	0	0	1	0	...	0
1	Alabama	1992	0	0	0	0	0	0	1	0	...	0
2	Alabama	1993	0	0	0	0	0	0	1	0	...	0
3	Alabama	1994	0	0	0	0	0	0	1	0	...	0
4	Alabama	1995	0	0	0	0	0	0	1	0	...	0
...	...	...	...	...	...	...	...	...	...	...	...	...
1495	Wyoming	2016	1	0	0	0	0	0	0	0	...	0
1496	Wyoming	2017	1	0	0	0	0	0	0	0	...	0
1497	Wyoming	2018	1	0	0	0	0	0	0	0	...	0
1498	Wyoming	2019	1	0	0	0	0	0	0	0	...	0
1499	Wyoming	2020	1	0	0	0	0	0	0	0	...	0

1500 rows × 137 columns

- ## NIBRS Data:

  It was a JASON file

  Its shape is only 4104 rows × 3 columns

Its dataframe:

	state	offense	response
0	HI	aggravated-assault	{'results': [{'count': 1364, 'data_year': 2018...
1	DE	aggravated-assault	{'results': [{'count': 3415, 'data_year': 2001...
2	PR	aggravated-assault	{'results': [], 'pagination': {'count': 0, 'pa...
3	TX	aggravated-assault	{'results': [{'count': 900, 'data_year': 1997}...
4	MA	aggravated-assault	{'results': [{'count': 54, 'data_year': 1994},...
...	...	...	...
4099	PA	all-offenses	{'results': [{'count': 6, 'data_year': 2013}, ...
4100	CT	all-offenses	{'results': [{'count': 77, 'data_year': 1998},...
4101	LA	all-offenses	{'results': [{'count': 537, 'data_year': 2003}...
4102	TN	all-offenses	{'results': [{'count': 394, 'data_year': 1997}...
4103	DC	all-offenses	{'results': [{'count': 91, 'data_year': 2000},...

4104 rows × 3 columns

- ## Revictim Data:

We have deleted the columns that contains NAN data

```
[ ]  Revictim_data=Revictim_data.dropna()
```

On this data we have used the operator 'groupby' to reduce the amount of data. 'groupby ' collects similar data or columns to each other, for example we can combine the columns of crimes that belong to the same category.

A groupby operation involves some combination of splitting the object, applying a function, and combining the results. This can be used to group large amounts of data and compute operations on these groups.

Groupby example:

```
offense_revicitm=Revictim_data[Revictim_data['recidivism_within_3years']==True].groupby("prison_offense").count()[["recidivism_within_3years"]]
offense_allrevicitm=Revictim_data.groupby("prison_offense").count()[["recidivism_within_3years"]]
```

- Groupby function of FBI data frame:

  We can combine similar data from

	state	offense	count	year	felony	invcommitment	invoutpatient	danger	drugmisdemeanor	alctreatment	...	expartedating
0	Hawaii	kidnapping-abduction	293	2018	1	1	1	1	1	1	...	1
1	Hawaii	robbery	1234	2018	1	1	1	1	1	1	...	1
2	Hawaii	sexual-assult-with-an-object	71	2018	1	1	1	1	1	1	...	1
3	Hawaii	assault-offenses	1364	2018	1	1	1	1	1	1	...	1
4	Hawaii	assault-offenses	1384	2018	1	1	1	1	1	1	...	1
...	...	...	...	...	...	...	...	...	...	...	...	...
6655	Illinois	assault-offenses	3	2001	1	1	0	1	1	0	...	0
6656	Indiana	assault-offenses	1	2013	0	0	0	0	0	0	...	0
6657	Indiana	assault-offenses	3	2013	0	0	0	0	0	0	...	0
6658	Washington	assault-offenses	1	2005	1	1	1	1	0	0	...	0
6659	Minnesota	homicide-offenses	1	1994	0	1	0	1	1	1	...	0

6660 rows × 139 columns

By State and year to deal with them together:

```
df_merged1=df_merged.groupby(['state','year']).sum()['count'
```

```
df_merged1
```

```
state      year
Alabama    1991    80436
           1992    68425
           2006     1044
           2007     1067
           2008     1040
                    ...
Wisconsin  2018    38147
           2019    40447
           2020    45214
Wyoming    2019      413
           2020     1119
Name: count, Length: 835, dtype: int64
```

- One Hot encoding to give the categorical feature values an understandable name meaning

```
] # Get the dummies

  one_hot = pd.get_dummies(Revictim_data_reg[['prior_conviction_episodes_1', 'prior_conviction_episodes_2',
         'prior_conviction_episodes_3', 'prior_conviction_episodes_4',
         'prior_conviction_episodes_5', 'prior_conviction_episodes_6',
         'prior_conviction_episodes_7','race','age_at_release','gang_affiliated']])
  # Drop columns as they is now encoded
  df_reg = Revictim_data_reg.copy().drop(columns=['prior_conviction_episodes_1', 'prior_conviction_episodes_2',
         'prior_conviction_episodes_3', 'prior_conviction_episodes_4',
         'prior_conviction_episodes_5', 'prior_conviction_episodes_6',
         'prior_conviction_episodes_7','race','age_at_release','gang_affiliated'])
  # Join the encoded df
  df_reg = df_reg.join(one_hot)


  Y=df_reg['supervision_risk_score_first']
```

- Dropping the feature of supervision_risk_score_first:

The structure of the project:

Flow Steps:

- Combining some crimes to be defined as violent
- Using API to send a request to load the data
- Encoding any Python object into JSON formatted by using the json.dumps () method
- Loading the JSON file into data variable
- The same encoding process was performed with the FBI JSON file
- Define some functions to be used within the code flow
- The starting of the Analysis work
- Performing Data Cleaning techniques
- Reading CSV file of the datasets of population_data, victim_data, and Revictim_data into dataframes
- Exploring the data frames by printing some examples or rows of them
- For victims data, we count of every incident category (victim _ demographic)
- Counting frequency of sex, age, and race crimes
- Visualizing the counts and plotting the frequency found for each type
- Comparing the relationship between the education level and rate of victimization
- Comparing the relationship between the income level and rate of victimization
- Citing the number of each crime or finding the count of each crime type
- Forming some plots to give insights about the relationships among features
- Finding the ratio of each crime to be compared with each other
- Working on Revictim data
- Dropping columns that contains NAN values
- Visualize the relationship between crime class and ratio_rectivim
- Visualize the relationship between the age victim and ratio
- Working on fire data
- Working on FBI data
- Grouping some features to visualize the relationship between them
- Examining the claim and hypotheses of the rate at which violence rate increases after 11/9 accident in USA
- Finding measurements needed to test the hypothesis, such as p value
- Building predictors and find the correlation between the features

Functions Description:

1-

```
def get_tasks(session):
    stat_off=[]
    tasks = []
    for crime in off:
      for state in sta :
        tasks.append(asyncio.create_task(session.get(url.format(crime,state), ssl=False)))
        stat_off.append()
    return tasks
```

This function is used to get tasks from a session. It takes in a session as an argument and creates two empty lists, stat_off and tasks. It then iterates through two lists, off and sta, and appends an asynchronous task to the tasks list using the session.get() method. Finally, it returns the tasks list.

2-

```
async def get_symbols():
    async with aiohttp.ClientSession() as session:
        tasks = get_tasks(session)
        # you could also do
        # tasks = [session.get(URL.format(symbol, API_KEY), ssl=False) for symbol in symbols]
        responses = await asyncio.gather(*tasks)
        print(responses)

        for response in responses:
             results.append(await response.json())
        with open("data3.json", 'w') as f:
            json.dump(results, f)

            return responses
```

This async function is used to get symbols from an API and store them in a json file. It uses the aiohttp library to create a ClientSession, which is then used to get tasks from the API. The tasks are then gathered using the asyncio library and stored in responses. The responses are then parsed into json format and stored in a file called data3.json. Finally, the responses are returned.

Bonus Task:

Steps:

- Data cleaning as mentioned
- Using `RandomForestClassifier model`
- Dropping NAN and missing values
- Finding correlation matrix
- Dropping unnecessary columns
- Splitting arrays or matrices into random train and test subsets
- 70 % training dataset and 30 % test datasets
- Creating a RF classifier
- Performing predictions on the test dataset
  - **ACCURACY OF THE MODEL: 0.7141848976711362**

The challenges/limitations/assumptions:

- Dealing with features name that are anonymous and their names do not reflect a suitable name
- Reading the dataset guidelines
- Dealing with missing data within the data frames