

Data link:

<https://www.kaggle.com/murderaccountability/homicide-reports>
(<https://www.kaggle.com/murderaccountability/homicide-reports>)

Sharable colab link:

https://colab.research.google.com/drive/19vFbBlmDCYbCuYvN4xHnHuaGYNAQ_HyE
(https://colab.research.google.com/drive/19vFbBlmDCYbCuYvN4xHnHuaGYNAQ_HyE)

Preparing packs and running GPU

In [0]:

```
#Loading Nvidia GPU extension. (Make sure that GPU hardware accelerator is enabled)
!nvidia-smi
```

Thu Nov 28 17:43:28 2019

```

+-----+
| NVIDIA-SMI 440.33.01      Driver Version: 418.67      CUDA Version: 10.1      |
+-----+-----+-----+
+-----+
| GPU  Name                Persistence-M| Bus-Id        Disp.A | Volatile Un
corr. ECC | Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  C
ompute M. |
+=====+=====+=====+
|    0   Tesla K80          Off      | 00000000:00:04.0 Off |
0 |
| N/A    59C    P0      61W / 149W |      363MiB / 11441MiB |      0%
Default |
+-----+-----+-----+
+-----+

+-----+
+-----+
| Processes:                                                         G
PU Memory |
| GPU      PID    Type    Process name                       U
sage      |
+=====+
=====|
+-----+
+-----+

```

In [0]:

```
#Loading packs

import pandas as pd
import numpy as np
import seaborn as sns
pd.set_option('display.max_colwidth', 0)

from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.preprocessing import sequence
from sklearn.model_selection import train_test_split

from keras.models import Sequential

from keras.layers import Dense, LSTM, Dropout, Activation, Dense, Input, CuDNNLSTM, Embedding, Dropout, SpatialDropout1D, Bidirectional
from keras.layers.normalization import BatchNormalization

from keras.initializers import Constant

from collections import Counter
from sklearn.feature_extraction import DictVectorizer
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import LabelEncoder
```

In [0]:

```
#Loading the Google Drive, since the files are located here
from google.colab import drive
drive.mount('/content/gdrive')
```

Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive", force_remount=True).

In [0]:

```
#Loading the dataset, this is read from a personal Google Drive folder. This might vary depending on save location
crimedata=pd.read_csv('gdrive/My Drive/SDS Pythongang/database.csv')
```

```
/usr/local/lib/python3.6/dist-packages/IPython/core/interactiveshell
1.py:2718: DtypeWarning: Columns (16) have mixed types. Specify dtype
option on import or set low_memory=False.
interactivity=interactivity, compiler=compiler, result=result)
```

In [0]:

```
#We look at the way the data looks, coloumns, rows and such:
crimedata.head(1)
```

Out[0]:

	Record ID	Agency Code	Agency Name	Agency Type	City	State	Year	Month	Incident	Crim
0	1	AK00101	Anchorage	Municipal Police	Anchorage	Alaska	1980	January	1	Mu Mansla

In [0]:

```
#Creating a copy of the crimedata for later.
#This will be used when we test the model against unsolved crime.
crimedata2 = crimedata.copy()
```

Cleaning, indexing and sampling the data

In [0]:

```
#First: We have a lot of data. Some of the cases are even unsolved.
#We remove these, since we will not be using them, when we are training the machine. The unsolved are spared for later.

crimedata_unsolved=crimedata["Crime Solved"]=="No"
crimedata.drop((crimedata[crimedata_unsolved].index), inplace=True)
```

In [0]:

```
#We also have a lot of missing information, so we also remove the instances, where this is the case.
#We specifically remove instances where there are missing information on Victim Race, Perpetrator Race, Weapon and Relationship.
crimedata_unknown_vic=crimedata["Victim Race"]=="Unknown"
crimedata.drop((crimedata[crimedata_unknown_vic].index), inplace=True)

crimedata_unknown_perp=crimedata["Perpetrator Race"]=="Unknown"
crimedata.drop((crimedata[crimedata_unknown_perp].index), inplace=True)

crimedata_unknown_weapon=crimedata["Weapon"]=="Unknown"
crimedata.drop((crimedata[crimedata_unknown_weapon].index), inplace=True)

crimedata_unknown_relation=crimedata["Relationship"]=="Unknown"
crimedata.drop((crimedata[crimedata_unknown_relation].index), inplace=True)
```

In [0]:

```
#We also have a lot of information, that we frankly - don't really need. So we also drop this.
```

```
crimedata_clean = crimedata.drop(["Crime Type", "Victim Ethnicity", "Perpetrator Ethnicity", "Agency Code", "Agency Name", "Agency Type", "Record Source", "Crime Solved", "Incident", "Victim Count", "Perpetrator Count"], 1)
crimedata_clean.head(1)
```

Out[0]:

	Record ID	City	State	Year	Month	Victim Sex	Victim Age	Victim Race	Perpetrator Sex	Per
0	1	Anchorage	Alaska	1980	January	Male	14	American/Alaska Native	Male	

In [0]:

```
#Now we check if there are any variables, that looks suspicious.
```

```
crimedata_clean.info(5)
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 335643 entries, 0 to 638453
Data columns (total 13 columns):
Record ID      335643 non-null int64
City           335643 non-null object
State          335643 non-null object
Year           335643 non-null int64
Month          335643 non-null object
Victim Sex     335643 non-null object
Victim Age     335643 non-null int64
Victim Race    335643 non-null object
Perpetrator Sex 335643 non-null object
Perpetrator Age 335643 non-null object
Perpetrator Race 335643 non-null object
Relationship    335643 non-null object
Weapon         335643 non-null object
dtypes: int64(3), object(10)
memory usage: 35.9+ MB
```

In [0]:

```
#Making a copy of the cleaned data for later - we will use this to showcase traditional machine learning later for comparison.
```

```
crimetestdata = crimedata_clean.copy()
```

In [0]:

```
#There are some numerics, but Perpetrator Age is listed as an object, even though it should be a numeric (e.g. 15) This is odd.  
#We therefor change this into an integer.
```

```
crimedata_clean["Perpetrator Age"] = pd.to_numeric(crimedata_clean["Perpetrator Age"], errors="coerce", downcast=None)
```

```
#To control, we now control The Perpetrator Age, it should now be of dtype int64.
```

```
crimedata_clean["Perpetrator Age"][:1]
```

Out[0]:

```
0      15
```

```
Name: Perpetrator Age, dtype: int64
```

In [0]:

```
#Now we limit the age in the data, since the lowest age of conviction in the US is in average about 6 years old.  
#So now we remove all instances, where the perpetrator has an age equal to or lower than 5.
```

```
crimedata_low=crimedata_clean["Perpetrator Age"]<=5  
crimedata_clean.drop((crimedata_clean[crimedata_low].index), inplace=True)
```

```
#We also filter out extreme highs. We think it should be fair to assume that 95+ is a wee bit high
```

```
crimedata_high=crimedata_clean["Perpetrator Age"]>=95  
crimedata_clean.drop((crimedata_clean[crimedata_high].index), inplace=True)
```

Indexing data

We couldn't get any other method to work, so we tried this This is a bit of a clunky solution, but it works.

In [0]:

```
#We group ages by certain values, this covers the following:
#6 - 12 = kid
#13 - 17 = teenager
#18 - 27 = young adult
#28 - 38 = adult
#39 - 59 = old adult
#60 - 95 = elderly
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(6
, "Kid")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(7
, "Kid")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(8
, "Kid")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(9
, "Kid")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(1
0, "Kid")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(1
1, "Kid")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(1
2, "Kid")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(1
3, "Teenager")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(1
4, "Teenager")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(1
5, "Teenager")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(1
6, "Teenager")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(1
7, "Teenager")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(1
8, "Young Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(1
9, "Young Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(2
0, "Young Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(2
1, "Young Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(2
2, "Young Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(2
3, "Young Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(2
4, "Young Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(2
5, "Young Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(2
6, "Young Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(2
7, "Young Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(2
8, "Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(2
9, "Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(3
0, "Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(3
1, "Adult")
```

```
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(3
2, "Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(3
3, "Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(3
4, "Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(3
5, "Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(3
6, "Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(3
7, "Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(3
8, "Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(3
9, "Old Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(4
0, "Old Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(4
1, "Old Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(4
2, "Old Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(4
3, "Old Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(4
4, "Old Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(4
5, "Old Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(4
6, "Old Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(4
7, "Old Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(4
8, "Old Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(4
9, "Old Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(5
0, "Old Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(5
1, "Old Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(5
2, "Old Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(5
3, "Old Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(5
4, "Old Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(5
5, "Old Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(5
6, "Old Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(5
7, "Old Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(5
8, "Old Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(5
9, "Old Adult")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(6
0, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(6
1, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(6
```

```
2, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(6
3, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(6
4, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(6
5, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(6
6, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(6
7, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(6
8, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(6
9, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(7
0, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(7
1, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(7
2, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(7
3, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(7
4, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(7
5, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(7
6, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(7
7, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(7
8, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(7
9, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(8
0, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(8
1, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(8
2, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(8
3, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(8
4, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(8
5, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(8
6, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(8
7, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(8
8, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(8
9, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(9
0, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(9
1, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(9
2, "Elderly")
```



```

crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(9
3, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(9
4, "Elderly")
crimedata_clean["Perpetrator Age"] = crimedata_clean["Perpetrator Age"].replace(9
5, "Elderly")

#Creating a filter for age column to show, that the code works.
filter = crimedata_clean["Perpetrator Age"] == "Kid"

#Printing only filtered columns
crimedata_clean.where(filter).dropna()

```

Out[0]:

	Record ID	City	State	Year	Month	Victim Sex	Victim Age	Victim Race	Pe
31	32.0	Juneau	Alaska	1980.0	June	Female	8.0	Asian/Pacific Islander	
4063	4064.0	San Diego	California	1980.0	August	Male	6.0	White	
4096	4097.0	San Diego	California	1980.0	November	Female	25.0	Black	
4370	4371.0	Santa Clara	California	1980.0	July	Male	27.0	White	
4655	4656.0	Yuba	California	1980.0	February	Male	76.0	Black	
...	
632651	632652.0	Vance	North Carolina	2014.0	July	Male	84.0	White	
634107	634108.0	Hamilton	Ohio	2014.0	July	Male	13.0	White	
634217	634218.0	Muskingum	Ohio	2014.0	June	Male	11.0	White	
634295	634296.0	Hamilton	Ohio	2014.0	March	Male	8.0	White	
637995	637996.0	Benton	Washington	2014.0	June	Female	16.0	White	

947 rows × 13 columns

In [0]:

```

#Making sure, that all variables have converted into strings with groupnames instead. There should be no numerics listed.
crimedata_clean["Perpetrator Age"].unique()

```

Out[0]:

```

array(['Teenager', 'Old Adult', 'Adult', 'Young Adult', 'Kid', 'Elderly'],
      dtype=object)

```

Sampling the data

One issue we have with the data, is the representation of the different races in the data. This is why we might have to sample the data instead, to make sure, that our machine does not get a bias towards heavily represented races.

In [0]:

```
#We quickly want to check the amount of murders by each race.
whitecrime = crimedata_clean["Perpetrator Race"].str.contains("White", case = False, na = False)
blackcrime = crimedata_clean["Perpetrator Race"].str.contains("Black", case = False, na = False)
nativecrime = crimedata_clean["Perpetrator Race"].str.contains("Native", case = False, na = False)
apcrime = crimedata_clean["Perpetrator Race"].str.contains("Asian/Pacific Islander", case = False, na = False)

print("The amount of murders committed by white perpetrators is ", len(crimedata_clean[whitecrime].index))
print("The amount of murders committed by black perpetrators is ", len(crimedata_clean[blackcrime].index))
print("The amount of murders committed by a perpetrator of native race is ", len(crimedata_clean[nativecrime].index))
print("The amount of murders committed by asian or pacific perpetrators is ", len(crimedata_clean[apcrime].index))
```

```
The amount of murders committed by white perpetrators is 166874
The amount of murders committed by black perpetrators is 150419
The amount of murders committed by a perpetrator of native race is 2
999
The amount of murders committed by asian or pacific perpetrators is
4646
```

In [0]:

```
#Ok, so we have a problem. Natives and Asians are represented way less in the statistics, than White and Black perpetrators.
#We do have more than 2.500 rows for each, which is why we can sample the data, to contain 2.500 samples of each race.

#Now we check the full names of the different races. This is needed for the next part, where we will pick a sample of 2.500 rows from each.
crimedata_clean["Perpetrator Race"].unique()
```

Out[0]:

```
array(['Native American/Alaska Native', 'White', 'Black',
      'Asian/Pacific Islander'], dtype=object)
```

In [0]:

```
#Now we can sample out 2.500 rows of each, and then merge them together.
```

```
whiteonly = crimedata_clean.loc[crimedata_clean["Perpetrator Race"]=="White"]
whitesample = whiteonly.sample(n=2500, random_state=42)
blackonly = crimedata_clean.loc[crimedata_clean["Perpetrator Race"]=="Black"]
blacksample = blackonly.sample(n=2500, random_state=42)
nativeonly = crimedata_clean.loc[crimedata_clean["Perpetrator Race"]=="Native American/Alaska Native"]
nativesample = nativeonly.sample(n=2500, random_state=42)
asianonly = crimedata_clean.loc[crimedata_clean["Perpetrator Race"]=="Asian/Pacific Islander"]
asiansample = asianonly.sample(n=2500, random_state=42)
```

```
#Merging the four tables
```

```
crimesample = whitesample.append([blacksample, nativesample, asiansample])
```

Data exploration

In [0]:

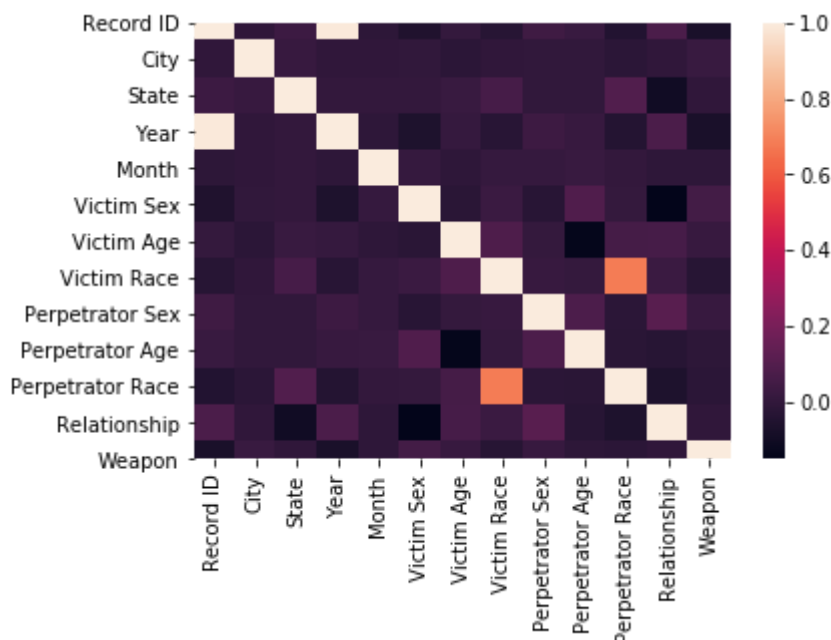
```
#To look at possible correlations, we try to plot all variables into a heatmap.
```

```
This is done to check for correlations.
```

```
crimedata_converted = crimesample.apply(LabelEncoder().fit_transform)
corr_vars = crimedata_converted.corr()
sns.heatmap(corr_vars)
```

Out[0]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff3ae0c1f28>
```



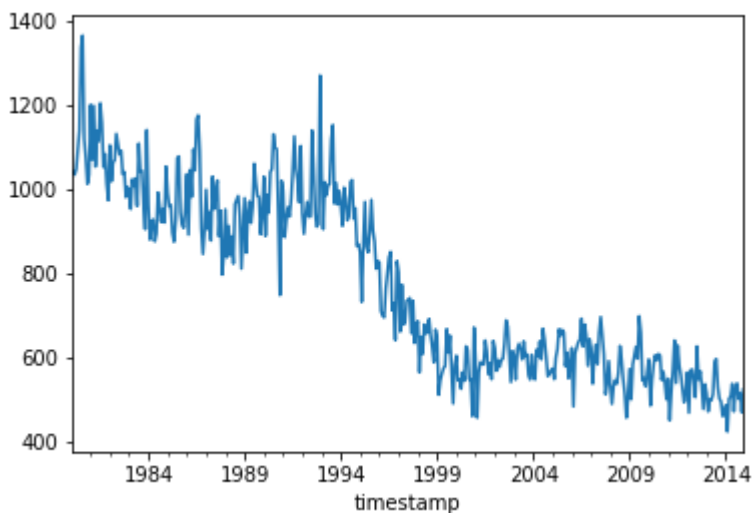
The heatmap above showed, that there is a strong correlation between Victim and Perpetrator Race. This was the same correlation, which we investigated in our assignment from M1. This time we want to look at more variables at the same time. We can also see some slight correlation between Victim Sex and Perpetrator Age, as well as Victim Race and Relationship.

In [0]:

```
#Looking at the development in murders throughout the years.
crimedata_clean_plot = crimedata_clean.copy()
date_cleaning = crimedata_clean_plot.Month.str.cat(crimedata_clean_plot.Year.astype(str), sep='-')
crimedata_clean_plot['timestamp'] = pd.to_datetime(date_cleaning)
crimedata_clean_plot.set_index('timestamp', inplace=True)
#Lower murders
crimedata_clean_plot[crimedata_clean_plot.index > pd.to_datetime('1-1-1980')].resample('M').size().plot()
```

Out[0]:

<matplotlib.axes._subplots.AxesSubplot at 0x7ff3a87d6908>



As we can see in the table above, there is a sudden dip in homicides. This is most likely due to the new abortion laws, which basically meant that the possible perpetrators were in fact never born. #FunFact
#TodayILearned

Table preperation for model training

In [0]:

```

#We now create a table containg information on the victim in 1 coloumn, and all
relevant information on the perp in seperate coloumnns.

#Creating a seperate table for the victim information. Grouped into one column.
This generates a "text" instead of a series of information.
victimdata = crimesample.assign(Victim_Information=crimesample["Month"].astype(s
tr) + ", " +
                                crimesample["Year"].astype(str) + " In the c
ity of " +
                                crimesample["City"].astype(str) + ", " +
                                crimesample["State"].astype(str) + " we made
a disturbing discovery as we found a dead " +
                                crimesample["Victim Race"].astype(str) + ",
" +
                                crimesample["Victim Sex"].astype(str) + ", "
+ "who seemed to be no older than " +
                                crimesample["Victim Age"].astype(str) + " ye
ars of age" + ", " + " we suspect, that the victim was killed with a " +
                                crimesample["Weapon"].astype(str))

Victim = pd.DataFrame(victimdata[{"Record ID", "Victim_Information"}])

#Creating a seperate table for the perpetrator information. This contains all th
e relevant variables regarding the perpetrator in one column
perpdata = crimesample.assign(Perp_Information=crimesample["Perpetrator Race"].a
stype(str) + ", " +
                                crimesample["Perpetrator Sex"].astype(str) +
", " +
                                crimesample["Perpetrator Age"].astype(str) +
", " +
                                crimesample["Relationship"].astype(str))

Perpetrator = pd.DataFrame(perpdata[{"Record ID", "Perp_Information",}])

```

In [0]:

```

#We merge the two tables, and drop the Record ID coloumn, since this serves no p
urpose other than for mergning.
policereport = pd.merge(Victim, Perpetrator, left_on= "Record ID", right_on="Rec
ord ID", how="left")
policereport = policereport.drop(["Record ID"], 1)
policereport.head(1)

```

Out[0]:

	Victim_Information	Perp_Information
0	July, 2003 In the city of Dallas, Texas we made a disturbing discovery as we found a dead White, Female, who seemed to be no older than 45 years of age, we suspect, that the victim was killed with a Shotgun	White, Male, Old Adult, Wife

In [0]:

```
#Code that counts occurrences of each word in the Perp_Information coloumn
policereport["perp_count"] = policereport["Perp_Information"].map(lambda x: dict(
    Counter(x.split(","))))

policereport.head()
```

Out[0]:

	Victim_Information	Perp_Information	perp_count
0	July, 2003 In the city of Dallas, Texas we made a disturbing discovery as we found a dead White, Female, who seemed to be no older than 45 years of age, we suspect, that the victim was killed with a Shotgun	White, Male, Old Adult, Wife	{'White': 1, 'Male': 1, 'Old Adult': 1, 'Wife': 1}
1	August, 1986 In the city of Hampden, Massachusetts we made a disturbing discovery as we found a dead White, Male, who seemed to be no older than 51 years of age, we suspect, that the victim was killed with a Blunt Object	White, Male, Young Adult, Stranger	{'White': 1, 'Male': 1, 'Young Adult': 1, 'Stranger': 1}
2	August, 1996 In the city of Cumberland, Pennsylvania we made a disturbing discovery as we found a dead Black, Male, who seemed to be no older than 18 years of age, we suspect, that the victim was killed with a Knife	White, Male, Adult, Acquaintance	{'White': 1, 'Male': 1, 'Adult': 1, 'Acquaintance': 1}
3	November, 1986 In the city of San Joaquin, California we made a disturbing discovery as we found a dead White, Male, who seemed to be no older than 34 years of age, we suspect, that the victim was killed with a Shotgun	White, Male, Adult, Friend	{'White': 1, 'Male': 1, 'Adult': 1, 'Friend': 1}
4	April, 1995 In the city of Bay, Florida we made a disturbing discovery as we found a dead White, Female, who seemed to be no older than 38 years of age, we suspect, that the victim was killed with a Handgun	White, Male, Old Adult, Wife	{'White': 1, 'Male': 1, 'Old Adult': 1, 'Wife': 1}

Preprocessing our data

In [0]:

```
#Initializing vectorizer and fitting
v = DictVectorizer()
v.fit(policereport["perp_count"])
#Creating y value
y = v.transform(policereport['perp_count'])
```

In [0]:

```
#We create vocabulary for the victim information
vocabulary_size= 1000
tokenizer = Tokenizer(num_words = vocabulary_size)
tokenizer.fit_on_texts(policereport["Victim_Information"])
sequences = tokenizer.texts_to_sequences(policereport["Victim_Information"])
```

In [0]:

```
#We create restrictions on the data and padding.  
MAXLEN = 50  
X = pad_sequences(sequences, maxlen=MAXLEN)  
indices=range(len(X))
```

In [0]:

```
#We create the test/training split  
X_train, X_test, y_train, y_test, index_train, index_test = train_test_split(X,  
y, indices, test_size=0.2)
```

Model creation and testing on our dataset

In [0]:

```
#We create my model  
  
embedding_vecor_length = 300  
  
model = Sequential()  
  
model.add(Embedding(vocabulary_size, embedding_vecor_length, input_length=MAXLEN  
))  
model.add(SpatialDropout1D(0.1))  
model.add(CuDNNLSTM(64))  
model.add(Dropout(0.2))  
model.add(BatchNormalization())  
model.add(Dense(64, activation="relu"))  
model.add(Dense(39, activation="sigmoid"))  
  
model.compile(loss="binary_crossentropy", optimizer="adam", metrics=["accuracy"  
])
```

In [0]:

```
#We test my model  
crimesolver = model.fit(X_train, y_train, epochs=20, batch_size=64, validation_s  
plit=0.1)
```


Train on 7200 samples, validate on 800 samples

Epoch 1/20

7200/7200 [=====] - 4s 556us/step - loss: 0.3068 - acc: 0.8724 - val_loss: 0.1951 - val_acc: 0.9279

Epoch 2/20

7200/7200 [=====] - 2s 299us/step - loss: 0.1891 - acc: 0.9288 - val_loss: 0.1857 - val_acc: 0.9283

Epoch 3/20

7200/7200 [=====] - 2s 300us/step - loss: 0.1827 - acc: 0.9303 - val_loss: 0.1818 - val_acc: 0.9288

Epoch 4/20

7200/7200 [=====] - 2s 301us/step - loss: 0.1788 - acc: 0.9313 - val_loss: 0.1803 - val_acc: 0.9298

Epoch 5/20

7200/7200 [=====] - 2s 296us/step - loss: 0.1749 - acc: 0.9320 - val_loss: 0.1804 - val_acc: 0.9278

Epoch 6/20

7200/7200 [=====] - 2s 297us/step - loss: 0.1722 - acc: 0.9330 - val_loss: 0.1818 - val_acc: 0.9291

Epoch 7/20

7200/7200 [=====] - 2s 296us/step - loss: 0.1700 - acc: 0.9335 - val_loss: 0.1750 - val_acc: 0.9326

Epoch 8/20

7200/7200 [=====] - 2s 303us/step - loss: 0.1674 - acc: 0.9340 - val_loss: 0.1743 - val_acc: 0.9307

Epoch 9/20

7200/7200 [=====] - 2s 297us/step - loss: 0.1659 - acc: 0.9346 - val_loss: 0.1748 - val_acc: 0.9300

Epoch 10/20

7200/7200 [=====] - 2s 300us/step - loss: 0.1635 - acc: 0.9350 - val_loss: 0.1728 - val_acc: 0.9312

Epoch 11/20

7200/7200 [=====] - 2s 300us/step - loss: 0.1620 - acc: 0.9353 - val_loss: 0.1699 - val_acc: 0.9318

Epoch 12/20

7200/7200 [=====] - 2s 300us/step - loss: 0.1603 - acc: 0.9358 - val_loss: 0.1720 - val_acc: 0.9318

Epoch 13/20

7200/7200 [=====] - 2s 297us/step - loss: 0.1597 - acc: 0.9362 - val_loss: 0.1730 - val_acc: 0.9303

Epoch 14/20

7200/7200 [=====] - 2s 299us/step - loss: 0.1581 - acc: 0.9363 - val_loss: 0.1725 - val_acc: 0.9318

Epoch 15/20

7200/7200 [=====] - 2s 296us/step - loss: 0.1572 - acc: 0.9366 - val_loss: 0.1722 - val_acc: 0.9325

Epoch 16/20

7200/7200 [=====] - 2s 295us/step - loss: 0.1558 - acc: 0.9372 - val_loss: 0.1742 - val_acc: 0.9311

Epoch 17/20

7200/7200 [=====] - 2s 299us/step - loss: 0.1550 - acc: 0.9375 - val_loss: 0.1746 - val_acc: 0.9294

Epoch 18/20

7200/7200 [=====] - 2s 296us/step - loss: 0.1541 - acc: 0.9379 - val_loss: 0.1748 - val_acc: 0.9305

Epoch 19/20

7200/7200 [=====] - 2s 297us/step - loss: 0.1533 - acc: 0.9381 - val_loss: 0.1730 - val_acc: 0.9322

Epoch 20/20

7200/7200 [=====] - 2s 297us/step - loss: 0.1528 - acc: 0.9384 - val_loss: 0.1722 - val_acc: 0.9308

Reviewing results

In [0]:

```
#How many results do we want shown from our model.
start = 0
end = 5
```

In [0]:

```
#Predicting on the test set.

y_pred = model.predict(X_test[start:end])
y_pred = np.where(y_pred > 0.3, 1, 0)
y_pred = [list(x.keys()) for x in v.inverse_transform(y_pred)]
y_test_s = [list(x.keys()) for x in v.inverse_transform(y_test[start:end])]

descr = policereport['Victim_Information'][index_test[start:end]]

#Comparing prediction to real results
pd.DataFrame({'prediction':y_pred, 'real': y_test_s, 'victim_Information': descr
})
```

Out[0]:

	prediction	real	victim_Information
2985	[Acquaintance, Male, Old Adult, Black]	[Adult, Male, Stranger, Black]	June, 2004 In the city of Shelby, Tennessee we made a disturbing discovery as we found a dead Black, Male, who seemed to be no older than 40 years of age, we suspect, that the victim was killed with a Handgun
1466	[Elderly, Male, Old Adult, Wife, White]	[Elderly, Girlfriend, Male, White]	July, 2007 In the city of Hillsborough, New Hampshire we made a disturbing discovery as we found a dead White, Female, who seemed to be no older than 62 years of age, we suspect, that the victim was killed with a Handgun
7483	[Acquaintance, Male, Young Adult, Native American/Alaska Native, White]	[Acquaintance, Male, Young Adult, Native American/Alaska Native]	January, 1992 In the city of Luzerne, Pennsylvania we made a disturbing discovery as we found a dead White, Male, who seemed to be no older than 18 years of age, we suspect, that the victim was killed with a Rifle
1246	[Elderly, Male, Old Adult, Wife, White]	[Male, Old Adult, Wife, White]	February, 1991 In the city of Galveston, Texas we made a disturbing discovery as we found a dead White, Female, who seemed to be no older than 39 years of age, we suspect, that the victim was killed with a Handgun
1888	[Acquaintance, Adult, Male, Young Adult, White]	[Acquaintance, Adult, Male, White]	October, 1986 In the city of York, Virginia we made a disturbing discovery as we found a dead White, Male, who seemed to be no older than 24 years of age, we suspect, that the victim was killed with a Knife

Is we can see, the model is pretty accurate. But with the relationship it often encompasses multiple forms of relationship. This is probably to ensure succes. The same goes for age.

Predicting on unsolved crime

In [0]:

#We use the previous created copy to look at unsolved cases.

```

crimedata_solved=crimedata2["Crime Solved"]=="Yes"
crimedata2.drop((crimedata2[crimedata_solved].index), inplace=True)
crimedata2.head(1)

```

Out[0]:

Record ID	Agency Code	Agency Name	Agency Type	City	State	Year	Month	Incident	Crime
2	3	AK00101	Anchorage Municipal Police	Anchorage	Alaska	1980	March	2	Murder Manslaughter

In [0]:

#We remove unknown information that is required to run our model.

```

crimedata_unknown_vic=crimedata2["Victim Race"]=="Unknown"
crimedata2.drop((crimedata2[crimedata_unknown_vic].index), inplace=True)

crimedata_unknown_weapon=crimedata2["Weapon"]=="Unknown"
crimedata2.drop((crimedata2[crimedata_unknown_weapon].index), inplace=True)

```

In [0]:

#Specifying the unknowns

```

crimedata2["Perpetrator Race"] = crimedata2["Perpetrator Race"].replace("Unknown", "Unknown Race")
crimedata2["Perpetrator Sex"] = crimedata2["Perpetrator Sex"].replace("Unknown", "Unknown Sex")
crimedata2["Relationship"] = crimedata2["Relationship"].replace("Unknown", "Unknown Relationship")

```

In [0]:

#Securing that the age variable is an integer.

```

crimedata2["Perpetrator Age"] = pd.to_numeric(crimedata2["Perpetrator Age"], errors="coerce", downcast=None)

```

Age indexing

In [0]:

```
#We group ages by certain values, this covers the following:
```

```
#0 - 5 = Unknown
#6 - 12 = kid
#13 - 17 = teenager
#18 - 27 = young adult
#28 - 38 = adult
#39 - 59 = old adult
#60 - 95 = elderly
#96 - 99 = Unknown
```

```
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(0, "Unknown")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(1, "Unknown")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(2, "Unknown")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(3, "Unknown")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(4, "Unknown")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(5, "Unknown")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(6, "Kid")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(7, "Kid")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(8, "Kid")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(9, "Kid")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(10, "Kid")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(11, "Kid")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(12, "Kid")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(13, "Teenager")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(14, "Teenager")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(15, "Teenager")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(16, "Teenager")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(17, "Teenager")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(18, "Young Adult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(19, "Young Adult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(20, "Young Adult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(21, "Young Adult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(22, "Young Adult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(23, "Young Adult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(24, "Young Adult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(25, "Young Adult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(26, "Young Adult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(27, "Young Adult")
```

```
Adult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(28, "Adult")
)
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(29, "Adult")
)
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(30, "Adult")
)
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(31, "Adult")
)
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(32, "Adult")
)
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(33, "Adult")
)
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(34, "Adult")
)
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(35, "Adult")
)
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(36, "Adult")
)
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(37, "Adult")
)
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(38, "Adult")
)
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(39, "Old Ad
ult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(40, "Old Ad
ult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(41, "Old Ad
ult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(42, "Old Ad
ult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(43, "Old Ad
ult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(44, "Old Ad
ult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(45, "Old Ad
ult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(46, "Old Ad
ult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(47, "Old Ad
ult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(48, "Old Ad
ult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(49, "Old Ad
ult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(50, "Old Ad
ult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(51, "Old Ad
ult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(52, "Old Ad
ult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(53, "Old Ad
ult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(54, "Old Ad
ult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(55, "Old Ad
ult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(56, "Old Ad
ult")
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(57, "Old Ad
ult")
```

```
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(58, "Old Ad  
ult")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(59, "Old Ad  
ult")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(60, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(61, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(62, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(63, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(64, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(65, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(66, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(67, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(68, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(69, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(70, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(71, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(72, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(73, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(74, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(75, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(76, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(77, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(78, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(79, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(80, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(81, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(82, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(83, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(84, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(85, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(86, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(87, "Elderl  
y")  
crimedata2["Perpetrator Age"] = crimedata2["Perpetrator Age"].replace(88, "Elderl
```

```

y")
crimedata2["Perpetrator Age"]= crimedata2["Perpetrator Age"].replace(89, "Elderl
y")
crimedata2["Perpetrator Age"]= crimedata2["Perpetrator Age"].replace(90, "Elderl
y")
crimedata2["Perpetrator Age"]= crimedata2["Perpetrator Age"].replace(91, "Elderl
y")
crimedata2["Perpetrator Age"]= crimedata2["Perpetrator Age"].replace(92, "Elderl
y")
crimedata2["Perpetrator Age"]= crimedata2["Perpetrator Age"].replace(93, "Elderl
y")
crimedata2["Perpetrator Age"]= crimedata2["Perpetrator Age"].replace(94, "Elderl
y")
crimedata2["Perpetrator Age"]= crimedata2["Perpetrator Age"].replace(95, "Elderl
y")
crimedata2["Perpetrator Age"]= crimedata2["Perpetrator Age"].replace(96, "Unknow
n")
crimedata2["Perpetrator Age"]= crimedata2["Perpetrator Age"].replace(97, "Unknow
n")
crimedata2["Perpetrator Age"]= crimedata2["Perpetrator Age"].replace(98, "Unknow
n")
crimedata2["Perpetrator Age"]= crimedata2["Perpetrator Age"].replace(99, "Unknow
n")
crimedata2["Perpetrator Age"]= crimedata2["Perpetrator Age"].replace("nan", "Unk
nown")

# creating a filter for age column
filter = crimedata2["Perpetrator Age"]=="Kid"

# printing only filtered columns
crimedata2.where(filter).dropna()

```

Out[0]:

	Record ID	Agency Code	Agency Name	Agency Type	City	State	Year	Month	In
120803	120804.0	WI07203	Wisconsin Rapids	Municipal Police	Wood	Wisconsin	1985.0	July	
449298	449299.0	AL00102	Birmingham	Municipal Police	Jefferson	Alabama	2003.0	December	
580040	580041.0	AR06002	Little Rock	Municipal Police	Pulaski	Arkansas	2011.0	November	
610425	610426.0	AR06002	Little Rock	Municipal Police	Pulaski	Arkansas	2012.0	July	

Making table for unknown crime

In [0]:

```
#We now create a table containg information on the victim in 1 coloumn, and all relevant information on the perp in seperate coloumns.

#Creating a seperate table for the victim information. Grouped into one column. This generates a "text" instead of a series of information.

unsolvedcrime = crimedata2.drop(["Crime Type", "Victim Ethnicity", "Perpetrator Ethnicity", "Agency Code", "Agency Name", "Agency Type", "Record Source", "Crime Solved", "Incident", "Victim Count", "Perpetrator Count"], 1)

victimdata2 = unsolvedcrime.assign(Victim_Information=crimedata2["Month"].astype(str) + ", " +
                                   crimedata2["Year"].astype(str) + " In the ci
ty of " +
                                   crimedata2["City"].astype(str) + ", " +
                                   crimedata2["State"].astype(str) + " we made
a disturbing discovery as we found a dead " +
                                   crimedata2["Victim Race"].astype(str) + ", "
+
                                   crimedata2["Victim Sex"].astype(str) + ", "
+ "who seemed to be no older than " +
                                   crimedata2["Victim Age"].astype(str) + " yea
rs of age" + ", " + " we suspect, that the victim was killed with a " +
                                   crimedata2["Weapon"].astype(str))

unsolvedvictim = pd.DataFrame(victimdata2[{"Record ID", "Victim_Information"}])

#Creating a seperate table for the perpetrator information. This contains all the relevant variables regarding the perpetrator in one column

perpdata2 = unsolvedcrime.assign(Perp_Information=unsolvedcrime["Perpetrator Race"].astype(str) + ", " +
                                   unsolvedcrime["Perpetrator Sex"].astype(str)
+ ", " +
                                   unsolvedcrime["Perpetrator Age"].astype(str)
+ ", " +
                                   unsolvedcrime["Relationship"].astype(str))

upp = pd.DataFrame(perpdata2[{"Record ID", "Perp_Information",}])
```

In [0]:

```
#Merging tables

pythomcruise = pd.merge(unsolvedvictim, upp, left_on= "Record ID", right_on="Record ID", how="left")
pythomcruise = pythomcruise.drop(["Record ID"], 1)

pythomcruise["perp_count"] = pythomcruise["Perp_Information"].map(lambda x: dict(Counter(x.split(","))))
```


In [0]:

```
pythomcruise.head(1)
```

Out[0]:

	Victim_Information	Perp_Information	perp_count
0	June, 1980 In the city of Anchorage, Alaska we made a disturbing discovery as we found a dead White, Male, who seemed to be no older than 32 years of age, we suspect, that the victim was killed with a Firearm	Unknown Race, Unknown Sex, Unknown, Unknown Relationship	{'Unknown Race': 1, ' Unknown Sex': 1, ' Unknown': 1, ' Unknown Relationship': 1}

Preprocessing

In [0]:

```
#Again, we create our tokenizer to understand the text.
```

```
tokenizer2 = Tokenizer(num_words = vocabulary_size)
tokenizer2.fit_on_texts(pythomcruise["Victim_Information"])
sequences2 = tokenizer2.texts_to_sequences(pythomcruise["Victim_Information"])
```

In [0]:

```
#Padding and restricting
```

```
MAXLEN = 50
X2 = pad_sequences(sequences2, maxlen=MAXLEN)
indices2=range(len(X2))
```

Reviewing results

In [0]:

```
#Predicting on the unsolved set.

z_pred = model.predict(X2[start:end])
z_pred = np.where(z_pred > 0.3, 1, 0)
z_pred = [list(x.keys()) for x in v.inverse_transform(z_pred)]
z_test_s = [list(x.keys()) for x in v.inverse_transform(y_test[start:end])]

descr2 = pythomcruise['Victim_Information'][index_test[start:end]]

pd.DataFrame({'Prediction': z_pred, 'Victim_Information': descr2})
```

Out[0]:

	Prediction	Victim_Information
2985	[Acquaintance, Male, Young Adult, Native American/Alaska Native]	November, 1980 In the city of Oakland, Michigan we made a disturbing discovery as we found a dead White, Male, who seemed to be no older than 26 years of age, we suspect, that the victim was killed with a Knife
1466	[Acquaintance, Adult, Male, Stranger, Young Adult, White]	September, 1980 In the city of District of Columbia, District of Columbia we made a disturbing discovery as we found a dead Black, Female, who seemed to be no older than 23 years of age, we suspect, that the victim was killed with a Strangulation
7483	[Adult, Male, Old Adult, Stranger, White]	December, 1981 In the city of Cook, Illinois we made a disturbing discovery as we found a dead Black, Male, who seemed to be no older than 62 years of age, we suspect, that the victim was killed with a Handgun
1246	[Acquaintance, Adult, Male, Young Adult, Native American/Alaska Native]	September, 1980 In the city of San Mateo, California we made a disturbing discovery as we found a dead White, Male, who seemed to be no older than 49 years of age, we suspect, that the victim was killed with a Blunt Object
1888	[Acquaintance, Adult, Male, Young Adult, Native American/Alaska Native]	June, 1980 In the city of Orange, Florida we made a disturbing discovery as we found a dead Black, Male, who seemed to be no older than 26 years of age, we suspect, that the victim was killed with a Handgun

In [0]:

```
#Controlling our results. Checking to see if the information that the model predicts on is correct. The number can vary depending on which row is being controlled.
```

```
pythomcruise[:1494]
```

Out[0]:

	Victim_Information	Perp_Information	perp_count
0	June, 1980 In the city of Anchorage, Alaska we made a disturbing discovery as we found a dead White, Male, who seemed to be no older than 32 years of age, we suspect, that the victim was killed with a Firearm	Unknown Race, Unknown Sex, Unknown, Unknown Relationship	{'Unknown Race': 1, ' Unknown Sex': 1, ' Unknown': 1, ' Unknown Relationship': 1}
1	August, 1980 In the city of Anchorage, Alaska we made a disturbing discovery as we found a dead White, Male, who seemed to be no older than 48 years of age, we suspect, that the victim was killed with a Handgun	Unknown Race, Unknown Sex, Unknown, Unknown Relationship	{'Unknown Race': 1, ' Unknown Sex': 1, ' Unknown': 1, ' Unknown Relationship': 1}
2	February, 1980 In the city of Juneau, Alaska we made a disturbing discovery as we found a dead Black, Male, who seemed to be no older than 27 years of age, we suspect, that the victim was killed with a Handgun	Unknown Race, Unknown Sex, Unknown, Unknown Relationship	{'Unknown Race': 1, ' Unknown Sex': 1, ' Unknown': 1, ' Unknown Relationship': 1}
3	July, 1980 In the city of Juneau, Alaska we made a disturbing discovery as we found a dead White, Male, who seemed to be no older than 33 years of age, we suspect, that the victim was killed with a Firearm	Unknown Race, Unknown Sex, Unknown, Unknown Relationship	{'Unknown Race': 1, ' Unknown Sex': 1, ' Unknown': 1, ' Unknown Relationship': 1}
4	October, 1980 In the city of Juneau, Alaska we made a disturbing discovery as we found a dead White, Female, who seemed to be no older than 22 years of age, we suspect, that the victim was killed with a Firearm	Unknown Race, Unknown Sex, Unknown, Unknown Relationship	{'Unknown Race': 1, ' Unknown Sex': 1, ' Unknown': 1, ' Unknown Relationship': 1}
...
1489	November, 1980 In the city of District of Columbia, District of Columbia we made a disturbing discovery as we found a dead Black, Male, who seemed to be no older than 26 years of age, we suspect, that the victim was killed with a Handgun	Unknown Race, Unknown Sex, Unknown, Unknown Relationship	{'Unknown Race': 1, ' Unknown Sex': 1, ' Unknown': 1, ' Unknown Relationship': 1}
1490	November, 1980 In the city of District of Columbia, District of Columbia we made a disturbing discovery as we found a dead White, Male, who seemed to be no older than 37 years of age, we suspect, that the victim was killed with a Handgun	Unknown Race, Unknown Sex, Unknown, Stranger	{'Unknown Race': 1, ' Unknown Sex': 1, ' Unknown': 1, ' Stranger': 1}
1491	November, 1980 In the city of District of Columbia, District of Columbia we made a disturbing discovery as we found a dead White, Female, who seemed to be no older than 81 years of age, we suspect, that the victim was killed with a Blunt Object	Unknown Race, Unknown Sex, Unknown, Unknown Relationship	{'Unknown Race': 1, ' Unknown Sex': 1, ' Unknown': 1, ' Unknown Relationship': 1}

	Victim_Information	Perp_Information	perp_count
1492	November, 1980 In the city of District of Columbia, District of Columbia we made a disturbing discovery as we found a dead White, Male, who seemed to be no older than 15 years of age, we suspect, that the victim was killed with a Handgun	Unknown Race, Unknown Sex, Unknown, Stranger	{'Unknown Race': 1, ' Unknown Sex': 1, ' Unknown': 1, ' Stranger': 1}
1493	November, 1980 In the city of New Castle, Delaware we made a disturbing discovery as we found a dead White, Male, who seemed to be no older than 47 years of age, we suspect, that the victim was killed with a Blunt Object	Unknown Race, Unknown Sex, Unknown, Unknown Relationship	{'Unknown Race': 1, ' Unknown Sex': 1, ' Unknown': 1, ' Unknown Relationship': 1}

1494 rows × 3 columns

TFIDF

Illustrating results on a simpler model

In [0]:

```
#Importing relevant packages for our TFIDF.

import itertools
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.metrics import accuracy_score, confusion_matrix

#We make our training and test set.
ytestvalue = crimetestdata["Perpetrator Race"]
x3_train,x3_test,y3_train,y3_test=train_test_split(crimetestdata["Victim Race"],
ytestvalue, test_size=0.2, random_state=42)

#We create a vectorizer.
tfidf_vectorizer=TfidfVectorizer(stop_words='english', max_df=0.7)

#We fit and transform the training set, and then we transform the test set.
tfidf_train=tfidf_vectorizer.fit_transform(x3_train)
tfidf_test=tfidf_vectorizer.transform(x3_test)
```

In [0]:

```
#Creating the model and testing the accuracy.

pac=PassiveAggressiveClassifier(max_iter=50)
pac.fit(tfidf_train,y3_train)

y3_pred=pac.predict(tfidf_test)
score=accuracy_score(y3_test,y3_pred)
print(f'Accuracy: {round(score*100,2)}%')
```

Accuracy: 87.33%

Conclusion

As we have shown, our deep learning model is superior to the more simple TFIDF model. For further explanation on the results, please see the stakeholder report.