**Machine learning project using Python**

**LINEAR REGRESSION MODEL ON THE ISSUE OF “SEXUAL HARRASMENT ON WOMEN IN INDIA”**

Palepu Tarun Sathwik-11812012

M.V.Sai Nethra-11812027

May 3,2020

**I. Definition**

**Project Overview**

A Woman is raped in India every 20 minutes. The infamous rape and murder case of Nirbhaya has heard the verdict after 8 years. India has made very little progress in preventing crimes against women. “Nothing has changed for women at all,” Asha Devi, Nirbhaya’s mother, angrily told the media following the court’s verdict on January 7. “So many of these incidents are continuing to happen.” Even the Supreme Court has admitted that changes effected to criminal law after the Nirbhaya case haven’t yielded results. In the recent past, gruesome cases of sexual assault have surfaced from all corners of the country, including the abduction, gang rape, and murder of a young lawyer in Jharkhand; the rape and murder of a 55-year-old cloth seller in Delhi’s Gulabi Bagh neighborhood; and a teenager in the state of Bihar who was gang raped and killed before being set ablaze.

“All the talk of the death penalty for rape just means we may be seeing more women murdered so they can’t remain alive as a witness,” says Janaki Vishwanathan, who works at a Delhi-based women’s self-help group. Making things worse is the low conviction rate in rape cases, one of the lowest in the world. Across India these figures were an abysmal 0.3 percent in 2018. According to the NCRB, 156,327 rape cases went on trial that year. Of these, trial was completed in only 17,313 cases and just 4,708 cases resulted in a conviction. There were acquittals in 11,133 rape cases and discharges in 1,472 cases.

Collecting all the data from data.gov.in and NCRB(National Crime Record Bureau)over the past 15 years made us realize there is a lot to educate India on matters of puberty and anti-rape agenda.so we have developed a statistical view of the numbers over the data collected from the last 15 years and trained a ML model to predict the next 5 years outcomes so that people will realize the seriousness of this atrocities against women in India.

**Problem Statement**

In lieu of the alarming situation in India, we decided to build a ML model and train the data over the last 15 years just to yield results for the next 5 years. surprisingly enough, the predictions and the graph analysis are almost accurate with the present statistics available on the internet. The project was made with better accuracy and minimum training and classification time.

**II. Analysis**

**Data exploration**

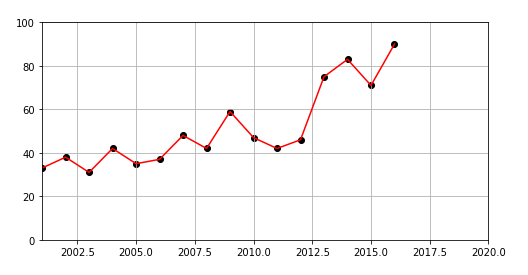
The datasets are downloaded from data.gov.in and NCRB(National Crime Records Bureau).we have collected datasets going way back to 2001 and till 2015.we have received the datasets in .excel formal and it listed all the types of harassment against women in India(organized by states) which included rape, attempt to rape,abused,etc.india has 29 states and the datasets included the data of 28 states till 2014. Telangana, which was formed in 2014 had to be considered as a separate state and hence considered null till 2014.All the States and Union-Territories are also being considered.

**Data cleaning**

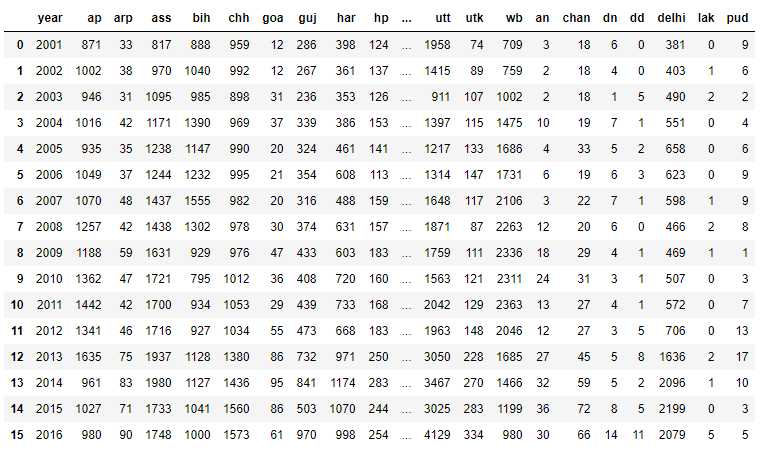
We got huge amounts of datasets from the government websites which had a lot of unnecessary data which had to be cleaned. Since our main focus is on rape and attempt to rape, we have cleaned the datasets pertaining to the topics outside of the requirement such as abuse, child abuse etc. Note that we have obtained the datasets from valid government sources and we are assuming the data provided is correct and valid (since there may be false accusation cases also and we chose to ignore that).

**II. Procedure**

We started with learning how to implement the linear model using various YouTube playlists. Then we first uploaded our datasets using pandas and stored in a file finaldata.csv. then we plotted graphs based on the dataset’s takin x-axis as the year and y-axis as the count as shown below:



Year vs count of cases of a state/UT



Finaldata.csv

Then the respective linear models were declared and each model was trained on the data in finaldata.csv. thus, we obtained our linear model which shows a predictable curve/line showing the trends till 2015 and above.

**II. Concept**

**Matplotlib**

Matplotlib is the most popular plotting library for Python. It was written by John D. Hunter in 2003 as a way of providing a plotting functionality similar to that of MATLAB, which at the time was the most popular programming language in academia.

Matplotlib offers a hierarchy of objects abstracting various elements of a plot. The hierarchy starts with the top-level Figure object that may contain a series of intermediate level objects and Axes – from Scatter, to Line and Marker, and all the way down to Canvas. In order to produce a plot on the screen, the matplotlib Figure instance must be coupled with one of the supported user interface backends such as TkInter, Qt, WxWidgets or MacOs. Outside of matplotlib documentation, user interface backends are typically referred to as “interactive”. In order to produce a file on a disk, matplotlib uses hardcopy backends for a variety of bitmap (png, jpg, gif) and vector (ps, ps, svg) file formats. Hardcopy backends are also called “non-interactive”.

A distinguishing feature of Matplotlib is the pyplot state machine which enables users to write concise procedural code. Pyplot determines the object to apply the relevant method from the context or creates the necessary objects on the fly, if they don’t exist. While this allows for fast experimentation, it can result in less reusable and less maintainable code.

In practice, it’s almost impossible to use matplotlib without pyplot. The Matplotlib user guide recommends using pyplot only to create figures and axes, and, once those are created, use their respective methods to create plots. This is reasonable, and we stick to this style in this tutorial, however I would advise not following it too rigidly when exploring new data. Having to look up which methods belong to which objects interrupts the flow of analytical thought and negatively affects productivity. The initial code can be easily converted to object-oriented style once you have finished exploring the data and know what visualizations you are going to need.

The ability to combine these two styles leads to great flexibility – according to the library maintainers, matplotlib makes easy things easy and hard things possible.

**When to use matplotlib**

The question is, what is hard and what is easy to implement in matplotlib?

There are two areas where matplotlib is particularly powerful:

* exploratory data analysis
* scientific plotting for publication

Matplotlib’s strength in exploratory data analysis comes from the pyplot interface. With pyplot you can generate a variety of plots with a small number of keystrokes and interactively augment existing figures with new data. Additionally, the seaborn library built on top of matplotlib provides even more visualizations with some basic data analysis, such as linear regression or kernel density estimation, built in.

The second area of matplotlib’s excellence is data visualization for publication. It can generate vector images in a variety of formats using its hardcopy (non-interactive) backends. When generating bitmap images matplotlib provides aesthetically pleasing rendering using Anti Grain Geometry (Agg). The default selection of axis annotations, fonts and ability to render mathematical notation using LaTeX syntax make it perfect for preparing figures for scientific journals or homework.

**When not to use matplotlib**

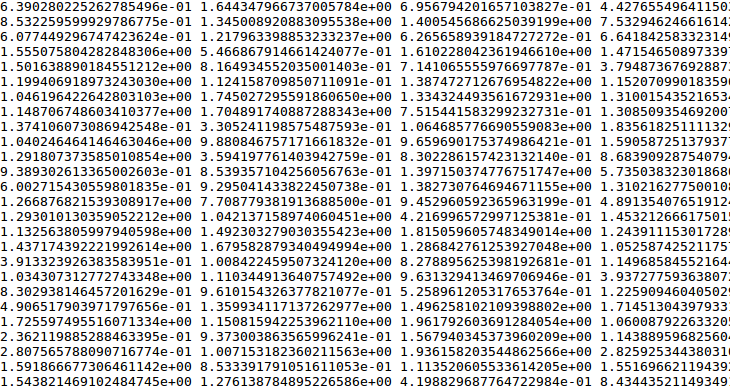
It’s true that you can create interactive graphical user interfaces with realtime updates using matplotlib. But from firsthand experience, I can vouch for a few other, better tools.

I would advise against using matplotlib for:

* Graphical user interfaces – instead, use pyforms.
* Interactive visualization for web – instead, use bokeh.
* Large datasets – instead, use vispy.
* Purpose of data visualization

The purpose of data visualization is to give us an insight into the data, so that we can understand it: we don’t understand the data when it’s just a pile of numbers.

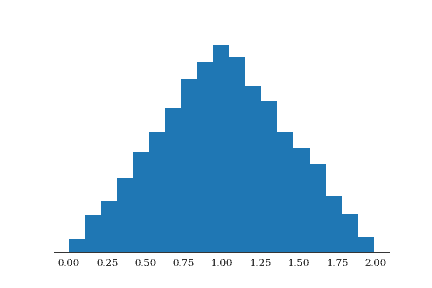
I see:



I understand: Nothing.

On the other hand, when we choose a proper visualization technique, the important things become clear.

I see:



I understand: It’s a triangle! (And the top is at 1.00)

It’s worth remembering that what we are after is insight during the entire visualization workflow – starting with data transformations, and ending with the choice of file format to save the images.

**Linear Regression Model**

Linear Regression is a supervised machine learning algorithm where the predicted output is continuous and has a constant slope. It’s used to predict values within a continuous range, (e.g. sales, price) rather than trying to classify them into categories (e.g. cat, dog). There are two main types:

**Simple regression:**

Simple linear regression uses traditional slope-intercept form, where m and b are the variables our algorithm will try to “learn” to produce the most accurate predictions. x represents our input data and y represents our prediction.

**y=mx+b**

**Multivariable regression:**

A more complex, multi-variable linear equation might look like this, where w represents the coefficients, or weights, our model will try to learn.

**f(x,y,z)=w1\*x+w2\*y+w3\*z**

The variables x,y,z represent the attributes, or distinct pieces of information, we have about each observation. For sales predictions, these attributes might include a company’s advertising spend on radio, TV, and newspapers.

**Sales=w1\*Radio+w2\*TV+w3\*News**

**Code to read finaldata.csv:**

import pandas as pd

data=pd.read\_csv("finaldata.csv")

data

**Code to plot the data available in finaldata.csv:**

from matplotlib import pyplot as plt1

plt1.rcParams["figure.figsize"] = (8, 4)

plt1.xlim(2001,2020)

plt1.ylim(0,6000)

plt1.scatter(data.year,data.ap,color="black")

plt1.plot(data.year,data.ap,color="violet”)

plt1.grid()

plt1.show()

**Code to implement linear regression model:**

from sklearn import linear\_model

model = linear\_model.LinearRegression()

**Polynomial regression:**

Polynomial Regression is a form of linear regression in which the relationship between the independent variable x and dependent variable y is modeled as an nth degree polynomial. Polynomial regression fits a nonlinear relationship between the value of x and the corresponding conditional mean of y, denoted E(y |x)

Why Polynomial Regression:

There are some relationships that a researcher will hypothesize is curvilinear. Clearly, such type of cases will include a polynomial term.

Inspection of residuals. If we try to fit a linear model to curved data, a scatter plot of residuals (Y axis) on the predictor (X axis) will have patches of many positive residuals in the middle. Hence in such situation it is not appropriate.

An assumption in usual multiple linear regression analysis is that all the independent variables are independent. In polynomial regression model, this assumption is not satisfied.

Uses of Polynomial Regression:

These are basically used to define or describe non-linear phenomenon such as:

* Growth rate of tissues.
* Progression of disease epidemics
* Distribution of carbon isotopes in lake sediments

The basic goal of regression analysis is to model the expected value of a dependent variable y in terms of the value of an independent variable x. In simple regression, we used following equation –

**y = a + bx + e**

Here y is dependent variable, a is y intercept, b is the slope and e is the error rate.

In many cases, this linear model will not work out For example if we analyzing the production of chemical synthesis in terms of temperature at which the synthesis take place in such cases we use quadratic model

**y = a + b1x + b2^2 + e**

Here y is dependent variable on x, a is y intercept and e is the error rate.

In general, we can model it for nth value.

**y = a + b1x + b2x^2 +....+ bnx^n**

Since regression function is linear in terms of unknown variables, hence these models are linear from the point of estimation.

**II. Conclusion**

In conclusion, this project has opened our minds into more of the social issues in India and how we, the youth of India can use technology and our education to bring a change in people on some of the alarming issues in India. We look forward to develop more projects that bring about a revolutionary change in India.

**Resources and References**

This project was implemented on anaconda navigator and Jupyter notebook.

<https://data.gov.in/catalog/district-wise-crimes-committed-against-women?filters%5Bfield_catalog_reference%5D=87613&format=json&offset=0&limit=6&sort%5Bcreated%5D=desc>

<https://ncrb.gov.in/crime-against-women-statesuts>

<https://www.indiatoday.in/india/story/ncrb-2018-woman-reports-rape-every-15-minutes-in-india-1635924-2020-01-11>

<https://www.youtube.com/watch?v=E5RjzSK0fvY>