CRIME AGAINST INDIAN WOMEN ANALYZER

A PROJECT REPORT

for

DATA MINING TECHNIQUES (SWE2009)

in

M.Tech. (Integrated) Software Engineering

by

SABRINA MANICKAM (19MIS0137)

LAASYA YARLAGADDA (19MIS0138)

SIRIGIRI SRI SAI SHARANYA (19MIS0266)

Under the Guidance of

Dr. SENTHILKUMAR N C

Associate Professor, SITE

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology and Engineering

April, 2022

DECLARATION BY THE CANDIDATE

I hereby declare that the project report entitled **“CRIME AGAINST INDIAN WOMEN ANALYZER”** submitted by me to Vellore Institute of Technology; Vellore in partial fulfillment of the requirement for the award of the course **Data Mining Techniques (SWE2009)** is a record of bonafide project work carried out by me under the guidance of **Dr. Senthilkumar N C.** I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other course.

Place : Vellore Signature

Date : 19-04-2022 Y. Laasya

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology & Engineering [SITE]

CERTIFICATE

This is to certify that the project report entitled **“CRIME AGAINST INDIAN WOMEN ANALYZER”** submitted by **Laasya Yarlagadda (19MIS0138)** to Vellore Institute of Technology, Vellore in partial fulfillment of the requirement for the award of the course **Data Mining Techniques (SWE2009)** is a record of bonafide work carried out by them under my guidance.

Dr. Senthilkumar N C

GUIDE

Associate Professor, SITE

Crime Against Indian Women Analyzer

Abstract

This paper presents different approaches to predict and classify crimes against women in India. Crime data was taken from Kaggle. After careful analysis of data using graphs, charts and heat maps, preprocessing steps like removal of unwanted rows and label encoding was carried out. Various algorithms like Decision Trees, Naïve Bayes, Support Vector Machine and Random Forest, BIRCH were implemented for classification, prediction and clustering. Further, various performance metrics like accuracy, precision, recall and f1 score were calculated. The experimental results were tabulated. Taking accuracy as the metric, Random Forest proved to be the best classifier and predictor with accuracy of 99.67% and 63.32% respectively.

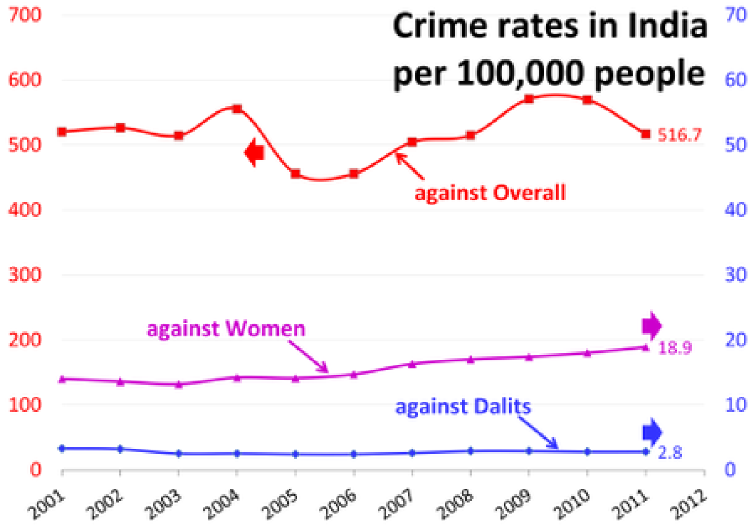
Keywords - Crime against women, Heat Maps, Decision Tress, Naïve Bayes, Support Vector Machine, Random Forest, Linear Regression, BRICH

I. INTRODUCTION

Crime Analysis includes investigating information about crimes to empower government authorities to more readily catch lawbreakers and forestall crimes. The manual procedures for anticipating crimes have been carried out for a really long time. The mining capacity, utilization of information and expanding force of knowledge innovation has expanded information assortment, storage and controls and are utilized to find comparative patterns. Data mining assists with aiding the investigation and forecast of assortments of perceptions of information. So it very well may be characterized as “Data Mining is the process of discovering new patterns from large data sets involving methods from statistics and artificial intelligence but also database management”.

II. BACKGROUND

Crime against women in India refers to any Violence against women in India. Crime actually more present than it may appear at first glance, as many expressions of violence are not considered crimes, or may otherwise go unreported or undocumented due to certain Indian cultural values and beliefs. According to the National Crime Records Bureau of India, reported incidents of crime against women increased 6.4% during 2012, and a crime against a woman is committed every three minutes. The security of the women is the utmost priority of any government in this world. In India, many policies and laws have been enforced to ensure the safety against women. Technology is being the biggest supporter to the government in this context. Data mining allows various techniques such as clustering classification; regression provides analysis in any form of data and helps intelligent predictions on the given dataset.



III. Literature Survey

A numerous Data mining algorithms have been discussed and proposed while dealing with crime data.

[1]. The paper proposed a work that collects and analyses the crime trend in Indian states and union territories by applying various classification techniques. Crime records for kidnapping, murder, rape, and dowry death were taken from the National Crime Records Bureau (NCRB) and the Open Government Data Platform. A training dataset (data from years 2001-12) with known class labels, was given to a predictive model that used the following classification techniques: Decision Trees, Naive Bayes, K- Nearest Neighbors, Random Forest, and AdaBoost (Adaptive boost). This prepared model was then applied to the test dataset (data from years

2013-14) and the results were evaluated by Accuracy, Precision, recall, and F-Measure.

[2]. This work presents a general algorithm for Spatio-Temporal Crime Prediction in developed cities by dividing the cities into subregions on the basis of crime dense regions. Crime data of wide areas of New York City and Chicago were used. The proposed algorithm first detects crime-dense regions done by geospatial clustering. The second step was spatial data splitting of the original crime data. The third step was aimed at extracting a specific crime prediction mode for each crime-dense region. A comparative analysis with regressive algorithms depicting that the proposed algorithm outperformed the other approaches.

[3]. This paper proposed a work to predict the occurrence of crimes for released prisoner propensity prediction. A new dataset with 30 attributes was created originally and exclusively created to define prisoners to understand them by their propensity to crime using psychology and behavioral factors. The research uses an analysis for seven search methods: Best first search, Greedy forward search, Evolutionary search, Wolf Search Method, Cuckoo Search algorithm, Flower pollination algorithm, and Ranker search algorithm. This was followed by a subset evaluation. Classification using methods like bagging with Random Forest, Decision tree naive Bayes, K Nearest Neighbors, etc were used. It is found that the wolf search algorithm, used with the correlation-based feature subset evaluation technique and radial basis function classifier, performs best providing 97.8% precision, 97.5% recall and low error values.

[4]. This paper proposed a multiple clustering approach based on fuzzy clustering theory to predict the high possibility of crime incidence by visualizing the crime analysis in various states in the US. The fuzzy c-means (FCM) algorithm works how an individual data point can be grouped in multiple clusters. This algorithm works by assigning membership to each data point corresponding to each cluster center on the basis of the distance between the cluster center and the data point. The input data has been taken from the US Arrests Database. After the model is built using the proposed process, predictions of US states that have most and least murder, assault, and rape arrests are done. The analysis is also visualized using various graphs like scatter plots, histograms, and bars.

[5]. The main goal through the paper was to propose a better clustering technique for criminal prediction to provide investigators with rich sources of information. Data collection, pattern identification, prediction, and visualization were applied in the papers. Primary data was collected from UCI and places like news, blogs, RSS Feeds, and social media. Simple K Means clustering was applied for the prediction. Visualization of suspected criminals was done using Graph Plotting.

[6]. The aim of this paper was to develop a system where the admincan enter crime details and the system will give some predicted output. The dataset considered is from the https://data.gov.in/ Indian government data source. The data collected is subjected to various data preprocessing techniques (removing noise, filling themissing values). The next step being data analysis where linear regression is used to explore various relationships among criminal activities and criminal attributes. The output of the analysis was a pie chart depicting the data for every state individually.

[7]. This paper worked to tackle the problems associated with the criminaljustice industry. So they considered textual corpus containing information about crime against women in India and to extract relations between the named entities present by a hierarchical graph-based clustering technique. Three types of entity pairs are PER- PER(person-person)which defines crime types like rape, murder etc, PER-LOC(person- location) describes the social status of the victim and ORG-PER(organization-person) Data is collected from newspapers like The Hindu, Times of India etc, from 2004 to 2016 and have been classified into the entities. After applying the proposed graph-based clustering algorithm, several clusters of named entity pairs are formed. External cluster evaluation techniques like Purity, Precision, Recall, F-measure and Random Index have been computed. Also few internal cluster evaluation indices like Score Function, Dunn, Davies-Bouldin, Silhouette, NIVA and Calinski-Harabasz are calculated. The Score Function gives the compactness of clusters. Larger the SF index implies better the clustersare. Higher value of the Dunn index represents good clustering. Different indices values are computed and detailed in the tables for both internal cluster indices and external clusterindices. The highest Purity score has been obtained for the PER-PER domain. Then a comparative study has been made among Infomap, Louvain, Girvan-Newman and Fast Greedy algorithms. The proposed clustering technique identifies significant crime patterns that can help both in criminology and the criminal justice industry.

[8]. In this paper the author has worked on analysis and prediction of different crimes in municipalities of Surigao del Norte.He used PHP DIDM database system to work on the dataset. For analyzing he used K-means algorithm for clustering theareas with number of recorded index and non-index crimes from 2013-2017 and result was implemented using KNIME (Konstanz Information Miner) analytics platform. ARIMA (1,0, 7) model was used to predict crimes from year 2018-2022. Analysis and prediction of each and every type of crime is done by dividing the municipalities into clusters, bar chartsand graphs are drawn for the same. Highest crime rate is recorded in every cluster for everycrime as well as prediction.

[9]. The paper starts with stating that they chose K-Means algorithmbecause of its simplicity and can also handle large datasets. In the methodology, they usedCRISP-DM model which contains 6 stages. Business Understanding Phase: Understandingthe research process as in objectives, needs and formulation of data mining problems. DataUnderstanding Phase: Data collection, identifying useful data and evaluating data quality. Data Preparation Phase: Data selection, Data preprocessing, Data transformation steps arecarried out at this phase. Data Modelling Phase: In this phase, data mining tools, algorithms used and modeling techniques are chosen and adjustments are made to obtain optimum results. Evaluation Phase: The models from the data modelling phase are evaluated based on Davies-Bouldin Index (DBI) and purity. Deployment Phase: In this phase, the reports and patterns are generated and those are presented in the form of Images and descriptions which can be easily understood. They also built and application using MySQL and PHP with the intent to help the Indonesian Government.

[10]. This paper aims at predicting crimes against women and criminalperformance in Tamil Nadu over 1500 crime records in around 20 districts. The researchershave tried to use clustering and classification algorithms for the same. Datasets were collected from the Tamil Nadu police department. All the necessary parameters have beenapplied in the Rapid Miner tool to extract data in K-Means Clustering Methods. Associationrules have been framed to predict and analyze the dataset. Crime type and criminal performance have been predicted from the Weka tool and group formation algorithm. Theresults have been tabulated and graphs of clustering have been attached.

[11]. This paper proposed an algorithm that forms fuzzy association rules between different crime rates. The aim of this study was to fill the blanks present in the current works on crime rate by efficiently being able to mine the reliable crime rate to crime rate relations from the access crime data. Data were taken from Chicago and NSW databases. After the pre-processing step the rate of crime was classified as high, normal, and low usingfuzzy association rules and hypothesis testing. Finally, a bridge between fuzzy transactionaldatasets and AR mining tools was built. Strong relations between rates of different crimescan be found in the results of case studies applied.

[12]. This paper worked on a new clustering algorithm to organize and retrieve information which is helpful in generating a charge sheet and a digital FIR by analyzing the complaint or the plain data given by the victim. It performs text mining on the complaint data using modified K-means clustering and SVM to automatically display articles in the digital charge sheet. This system can also be referred to by new officers to study how the different cases were solved previously.

[13]. This paper focused on detection and prediction of crime using crime reports. This method can also be used to track the criminals in a faster way. Crimes that are reported as fraud can also be detected. They defined every hotspot into crime patterns using clustering methods. Using Bayesian classifiers they tried to predict the crime at a specific area and specific time. For predicting the crime they needed crime month, crime day of theweek, crime time and location. Naive formula is used to find out the probability of the crime.

[14]. This paper proposed a system to analyze the crime data by recognizing the named entities and extracting relationships among them. The data is collected from the online versions of several newspapers and the crime-related data is extracted, which then undergoes data pre-processing techniques such as stop word removal, stemming and the named entities are recognized with the help of parts of speech tagging and NP chunking, to place them under the three domains (PER- PER), (PER– LOC), (PERORG) and NE pair similarity graph is formed which again is analyzed with the help of paraphrase collection to form sub clusters. The clusters formed are then evaluated by supervised and unsupervised techniques and also are tested for some existing classifiers like KNN, NB, SVM and Neural Network. Finally, the output given by this method provides insight on different characteristics of crime in India that took place over the past.

[15]. This paper used the K-means clustering technique to cluster crime against women into various sections based on the type of crime and the state it was perpetrated in. A dataset taken for the clustering analysis consists of mixed data of all the crimes across all the age groups in India. After data pre-processing and classification into different crimes against women, the Weka tool was for clustering analysis. The result obtained was a table giving various clusters and the included states along with graphical representations.

[16]. This paper worked on predictive analysis of crime dataset the Communities and crime from UCI repository. It consists of crime data in Chicago, a city with highest crime rate in US. The aim was to predict a new feature or attribute called Per Capita ViolentCrime.80% of data was used for training and 20% for testing. So, for this few predictive models like Decision Trees and Random Forest Classification, Naive Bayes Classification and Linear Regression have been used. For data preparation Pandas and Numpy have been installed. Information about the algorithms has been given. Entropy is used as criteria for splitting of branches. The following performance metrics have been calculated in every algorithm: Cross, Validation score, Accuracy, Precision, Recall, F1 score, Mean squared error. The data set was pre-processed and cleaned and a separate column for new attribute was made with a threshold value of 0.1. Now with this dataset experiments were done using all the mentioned algorithms on both cleaned data and uncleaned data. The results for every algorithm have been detailed in the paper along with top 10 features extracted. Thus, conclusion is that with Random Forest Classifier the results were balanced with respect to accuracy, precision, recall and F1 score. While Linear regression gave the lowest values in these performance metrics.

[17]. This paper has talked about different data mining algorithms and pre- processing steps to detect the crime patterns in the US country. They have started with data collection, data pre-processing, and data filtering and in the end linear regression. It also talks about Weka soft which is used for data analyzing. Visualization gives the final output. One advantage of this tool is that clustering is done automatically. Also comparison of crimes over a period is done using pie-charts. Linear regression is used to predict the number, let it be age, weight and so on.

[18]. This paper used time series analysis on Chicago crime data from 2017to May 2020. Aim was to understand the main use of time series algorithms to find the top crimes in Chicago on monthly, daily, weekly basis. Time series model of ARIMA modelis used to forecast top crimes in the city. The method contains four stages namely identification, estimation, diagnostics checking, and forecasting. This method helps in analyzing longitudinal data with a correlation among neighbouring observations. The data isextracted from Chicago Police Department's CLEAR system which includes 8 lakh crimeevents. The main goal of this research is to predict the crime events based on crime tookplace in past and to plan the activities for the prevention of crimes. Top 5 crimes of the cityin all the years have been graphed on monthly, weekly and daily basis. The results have showed that most of the crimes have been taken place in the months of May and April.The study presents 13 attributes in seven events like arrest, domestic violence, and theft and so on. Accuracy of existing ARIMA and Enhanced ARIMA model has been comparedas it shows that enhanced model has an accuracy of 91,4% which is greater than existing. Additional to accuracy the error rates are also measured using Mean Absolute Error (MAE),Mean Square Error (MSE) and Root of Mean Square Error (RMSE) and their results have been noted. This model can help the Chicago Police Department to predict the crimes in a better way.

[19]. This paper proposed a new approach using K-means clustering algorithm to reduce the complexity of crime investigation and predict the crime rate. This paper pointed out the limitations in the existing system like threatening of victim by criminals, a lot of manual work and so on. To overcome these limitations they have proposed a new approach based on K-means. Initially the system was trained with the criminal dataset including skin color, hair color, face type etc. Later the information about the criminal from the victim is collected and then a questionnaire is generated to trace the criminal which results in criminal view sheet. These details are checked in the dataset using K-means algorithm and identify the victim. Based on the result crime mapping is done and crime data is added to the crime dataset. Finally statistical report is generated. This leads to better communication, also reduces time and money.

[20]. This paper worked on crime analysis in states Of US using the crime dataset of FBI. The dataset had 14 attributes like states, city, population, violent crime, property crime and so on. The first step was data-preprocessing and cleansing. The size of the data was reduced and they applied two filters in WEKA tool one is Replace missing values and other is Numeric to nominal. The next step was applying data mining algorithms and they chose classification for their research. They used 6 classification algorithms namely J48 Pruned Tree, REP tree, Decision Table, JRIP rules, Random Tree, Naïve Bayes on WEKA tool. A comparison was made among the 6 algorithms based on time taken to build the model, correctly classified instances, incorrectly classified instances, Root mean squared error, TP rate, FP rate, Precision, Recall, F-Measure vales. A test mode 10-Fold Cross Validation has been used for evaluation purpose. It was shown that Reduced Error Pruning Tree and Naiive Bayes have produced better results when compared to other algorithms. Among the types of crimes mentioned, rate of property crime and rate of larceny-theft are high in the populated states. A detailed accuracy tale has been shown in the paper.

[21]. This paper focuses on implementing textual data analytics bycollecting the data from different news feeds and providing visualization. Data was taken from news feeds and databases of the Indian Government. The news feed data is convertedinto XML, to be in a machine-readable format. The proposed framework uses Naive Bayes for the classification of crimes into subcategories, Mallet Package for extracting keywords from news feeds. Further, the Kernel Density Estimation identifies the density of crime. Visual hotspots of crime in India are the output of the analysis. The ARIMA Model has been used to validate the proposed model and an equivalent prediction performance was found.

[22]. This paper proposed a schema-based crime incident description to improvethe efficiency of the crime prevention measures. The crime incident report: 1) describes thecharacteristics of a crime incident and its possible elements, and 2) offers a two-level offense classification scheme based on relevant criteria. In this study data taken from the NCRB (2001-2021) was pre-processed by converting timestamps to break year, months, day, hour, minute, and seconds. The dataset is further divided into two and trained using Random Forest and Decision Trees is applied. Crime type is predicted through a Graphical User Interface (GUI) by obtaining test feedback from the users. The final model predicts the type of crime depending upon the dataset when the necessary attributes are inputted.

[23]. This paper aims at examining associative mining procedures other than clustering in analyzing crime against women in and around Tamil Nadu. They have compared two different algorithms called Apriori and Expectation-Maximization for the same. Dataset was collected from Tamil Nadu police department. The target of utilizing Apriori calculation is to discover visit thing sets and relationship between various thing sets, affiliation imperative. This paper worked on rape and endeavor commit rape in the past four years 2015-2019.WEKA tool is used for pre-processing as well as evaluating data. The data file contained 8 attributes and 1350 occurrences. Demonstrations practices of both the methods have been shown in the tabular form using WEKA Explorer. It was proved that Apriori Algorithm and it's association rules produced better results than Expectation- Maximization Algorithm. WEKA boundary is an identical beneficial method permits to manipulator indicate numerous altered algorithms and associate them to influence the precisely prerequisite outcomes.

[24]. This paper that aims to expect which category of crime is most likely to take place at a detailed time and place in Calabar, Nigeria. The proposed model contains 3 main steps: 1. Data Collection 2. Pattern Identification 3. Prediction. The data is collected from various newspapers, websites, blogs, and social media. Different algorithms are used to train the dataset. For finding the crime pattern of a particular place many attributes of that place are considered and the Apriori algorithm is used. Various ensemble methods such as Bagging, AdaBoost, and ExtraTree Classifier classify the type of crime occurring based on time and location.

[25]. This paper proposes the use of time series ARIMA model to calculate crime. The data provided is five-year crime data (2012 to 2016) and the crime rate for 2 years (2017 to 2018). It also claimed that ARIMA model has higher fitting values with exponential smoothing. The dataset used was LC (London Crime) dataset and also data collected from various websites, blogs which is processed using Microsoft excel to convert it intostructured data. All the data is used for prediction with the help of IBM SPSS. Linear regression and ARIMA model were compared and ARIMA model was finalized. To verifythe model, they took data from 2012 to 2015 and predicted crime patterns for 2016, the actual and predicated values were very close to reality and the forecasted model gave 80% accurate results. Some statistical tables related were also presented.

[26]. This paper used various predictive and visualizing techniques to analyze data related to three different cities. Apart from data mining techniques (Data preprocessing, classification) various machine learning techniques (SVM, Random Forest, Neural Networks) were also used and the efficiency of the algorithms was tested. Using data mining entity extraction and clustering was performed and the result is used to perform linkanalysis to discover frequently occurring item sets. This paper concludes that the model along with a machine learning algorithm performs better than data mining.

[27]. This paper explains mainly three aspects regarding cyber security and safety.The first being the process of cyber-attack detection. In this step we will analyze and classify cyber incidents. In the second, forecasting upcoming cyber-attacks and controllingcyber terrorism. Third one is about the theoretical background and usability of AI with datamining approaches for addressing the above issues through detection and prediction. They have used data mining tools along with AI and ML methods to present visual interpretations related to cyber-crimes.

[28]. This paper used four different datasets and various classification algorithms in the givenpaper. The study is divided into two parts.

1. Applying four classification algorithms (KNN, Logistic Regression, Linear Regression, Random Forest Classifier) on the first three data sets (Los Angeles, Chicago, Egypt datasets). The result of this study is analyzed in terms of accuracy and it was noted that Random Forest Classifier achieved a higher score than others and also January month has higher number of crimes.

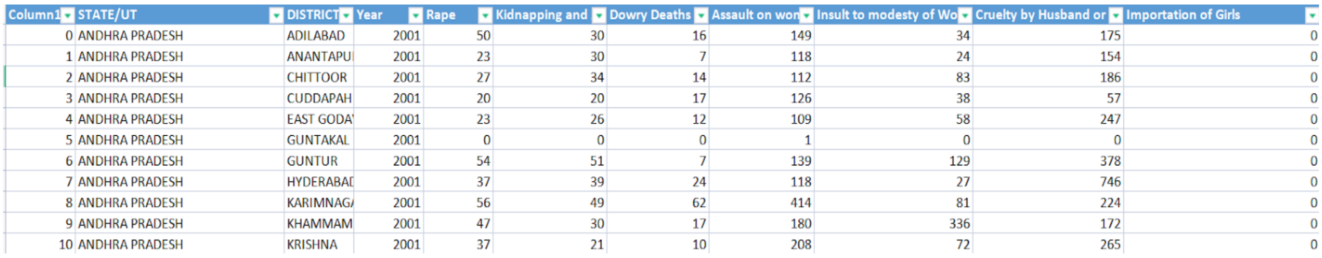
2. Applying five different classification algorithms (Decision trees, Gaussian NB, SVM, Gradient Boosting, Random Forest) to the United States dataset. The performance of the five classifier algorithms were analyzed in terms of accuracy, in terms of recall and Random Forest again achieved a higher score and in terms of precision Naïve Bayes topped.

[29]. This paper has focused on crime against women and presented a methodology to estimate and visualize the level of insecurity in geographical areas, using spatiotemporal analysis and data mining techniques. Data was taken from the Mexican Government Portal. The data structure was denormalized to contain the following domains: crime, year, date, hour, municipality, suburb, street, and in some field's latitude and longitude. The methodology was based on the CRISP-DM methodology. Further data cubes were built and an exploration-based level of risk was calculated. Various results suchas crime hotspots, percentage of violence against women, most used words in digital media,andmost dangerous day of the week were found.

[30]. This paper analyzed the crime related to rape in India using two text mining algorithms. The data set used is of rape cases in India collected from https://data.gov.in/. Various preprocessing techniques (Tokenization, Removal of stop word, Stemming) and text transformation techniques were used to transform the text into Vector space model and bag of words notation. Feature selection was used to select the relative features. On the resulting data, Firstly, classification with KNN to evaluate the relation between years and rape cases. The results said that the cases of rape are increasing as the year increases and Delhi state has the maximum number of rape cases. Secondly, K- Means Clustering algorithm was used which again gave the maximum percentage (16%) isshown by the Delhi Cluster.

IV. DATASET DESCRIPTION & SAMPLE DATA

Data used for this project has been taken from **Kaggle.** Originally the dataset contains 10677 rows and 11 columns. The attributes of the dataset are the Index, the State/UT, District, year, the crime types such as Rape, Kidnapping and Abduction, Dowry deaths, Assault on women, Insult to modesty, Cruelty by husband or his relatives, and Importation of girls. Out of the 11 attributes 2 attributes namely State and District were of type object while all others were 64 bit integers. The data contained no null values. In India there are 36 State/UT but the dataset had 72 States/UTs. This was because some States were mentioned in uppercase and some in lowercase. To correct this we converted all the states into lower case. While inspecting the data we found that some District entries were like “Total”, ”zz total”, so these rows were dropped from the dataset. Some analysis of the data was carried out using various charts and bar graphs.



Column1	STATE/UT	DISTRICT	Year	Rape	Kidnapping and	Dowry Deaths	Assault on won	Insult to modesty of Wo	Cruelty by Husband or	Importation of Girls
0	ANDHRA PRADESH	ADILABAD	2001	50	30	16	149	34	175	0
1	ANDHRA PRADESH	ANANTAPUI	2001	23	30	7	118	24	154	0
2	ANDHRA PRADESH	CHITTOOR	2001	27	34	14	112	83	186	0
3	ANDHRA PRADESH	CUDDAPAH	2001	20	20	17	126	38	57	0
4	ANDHRA PRADESH	EAST GODA'	2001	23	26	12	109	58	247	0
5	ANDHRA PRADESH	GUNTAKAL	2001	0	0	0	1	0	0	0
6	ANDHRA PRADESH	GUNTUR	2001	54	51	7	139	129	378	0
7	ANDHRA PRADESH	HYDERABAI	2001	37	39	24	118	27	746	0
8	ANDHRA PRADESH	KARIMNAG/	2001	56	49	62	414	81	224	0
9	ANDHRA PRADESH	KHAMMAM	2001	47	30	17	180	336	172	0
10	ANDHRA PRADESH	KRISHNA	2001	37	21	10	208	72	265	0

Creation of Target Variable

To identify areas as Low, Medium or High we have created a Target variable (Class label).

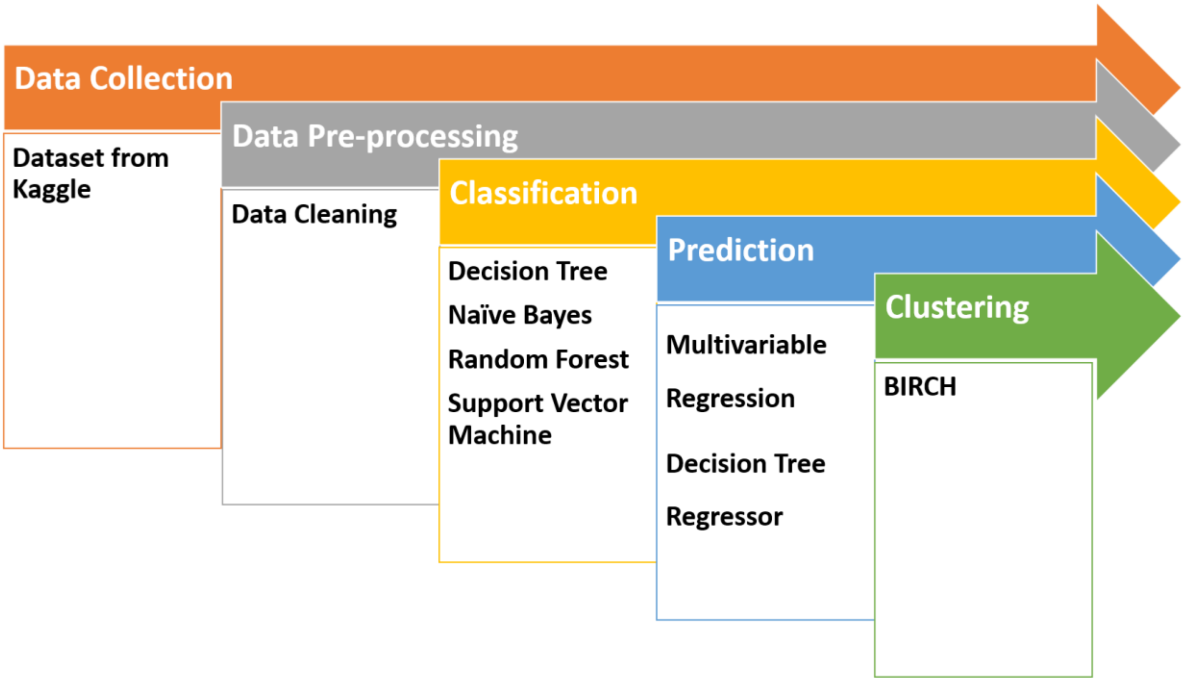
We assigned a rank order to all the crime attributes. There were:

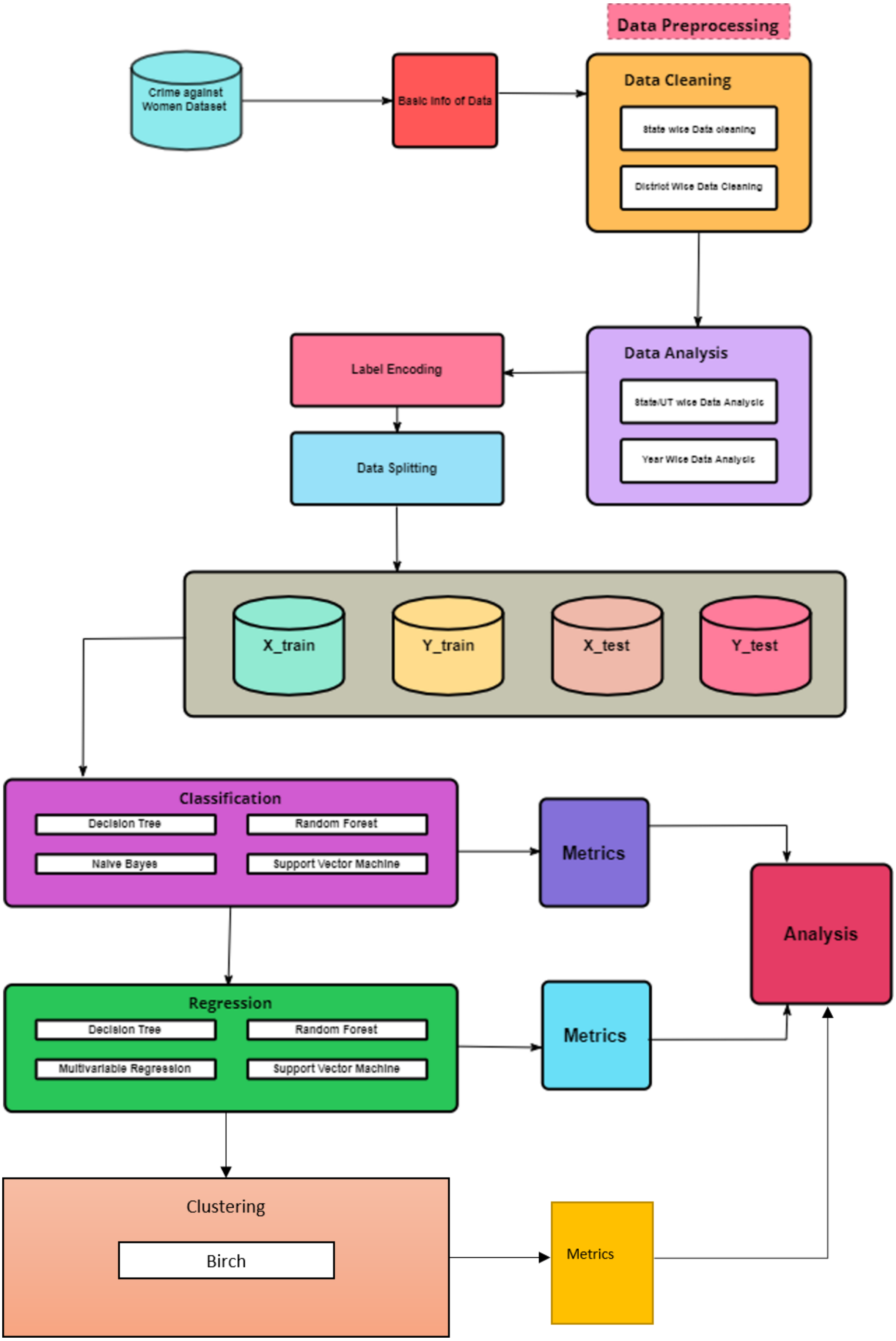
Crime Attribute	**Weightage**
Rape	7
Kidnapping and Abduction	6
Dowry Deaths	5
Assault on women with intent to outrage her modesty	4
Insult to modesty of Women	3
Cruelty by Husband or his Relatives	2
Importation of Girls	1

Total= [Rape value] *7+ [Kidnap and abduction value] *6+ [Dowry death value] *5+ [Assault value]* 4+[Insult value]* 3+[Cruelty value]* 2+[Importation value]*1

We label encoded the data into numbers using sklearn.preprocessing classLabelEncoder. Data was split in train and testing sets namely X_train and Y_train for training dataand X_test and Y_test for testing data using the train_test_split from the sklearn.model_selection.After splitting the training and testing, the shape of the data ((7130, 10), (3056, 10)).

V. PROPOSED ALGORITHM WITH FLOWCHART




Data Preprocessing
Crime against Women Dataset
Basic Info of Data
Data Cleaning
State wise Data cleaning
District Wise Data Cleaning
Data Analysis
State/UT wise Data Analysis
Year Wise Data Analysis
Label Encoding
Data Splitting
X_train
Y_train
X_test
Y_test
Classification
Decision Tree
Random Forest
Naive Bayes
Support Vector Machine
Metrics
Analysis
Regression
Decision Tree
Random Forest
Multivariable Regression
Support Vector Machine
Metrics
Clustering
Birch
Metrics

Classification

We applied the following algorithms:

1. Decision Tree

2. Random Forest

3. Naïve Bayes

4. Support Vector Machine

A Decision tree is a flowchart like tree structure, where each internal node denotesa test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

Random Forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e., every pair of features being classified is independent of each other.

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problemsin Machine Learning.

Prediction

We applied the following algorithms:

1. Decision Tree

2. Random Forest

3. Multiple Regression

Multiple regressions are a statistical technique that can be used to analyze the relationship between a single dependent variable and several independent variables. The objective of multiple regression analysis is to use the independent variables whose values are known to predict the value of the single dependent value.

Clustering

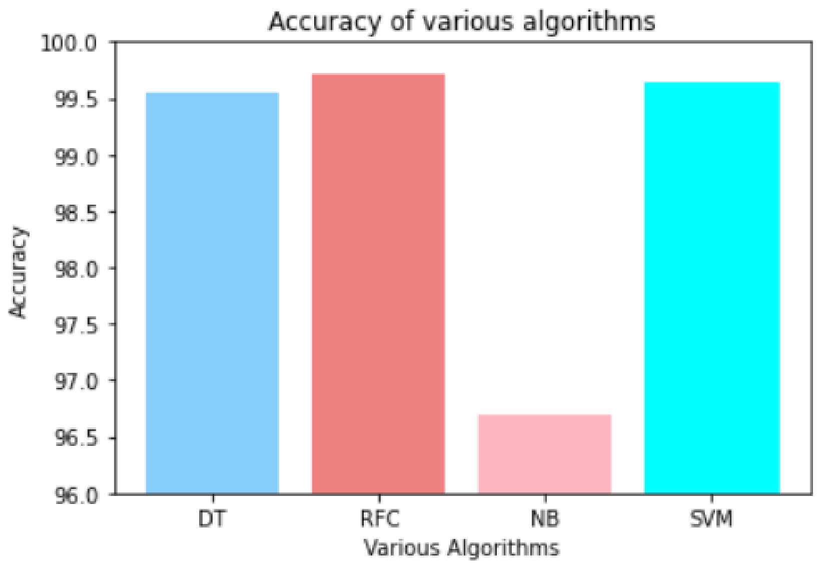
Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) is a clustering algorithm that can cluster large datasets by first generating a small and compact summary of the large dataset that retains as much information as possible. This smaller summary is then clustered instead of clustering the larger dataset. BIRCH is often used to complement other clustering algorithms by creating a summary of the dataset that the other clustering algorithm can now use. However, BIRCH has one major drawback – it can only process metric attributes. A metric attribute is any attribute whose values can be represented in Euclidean space i.e., no categorical attributes should be present.

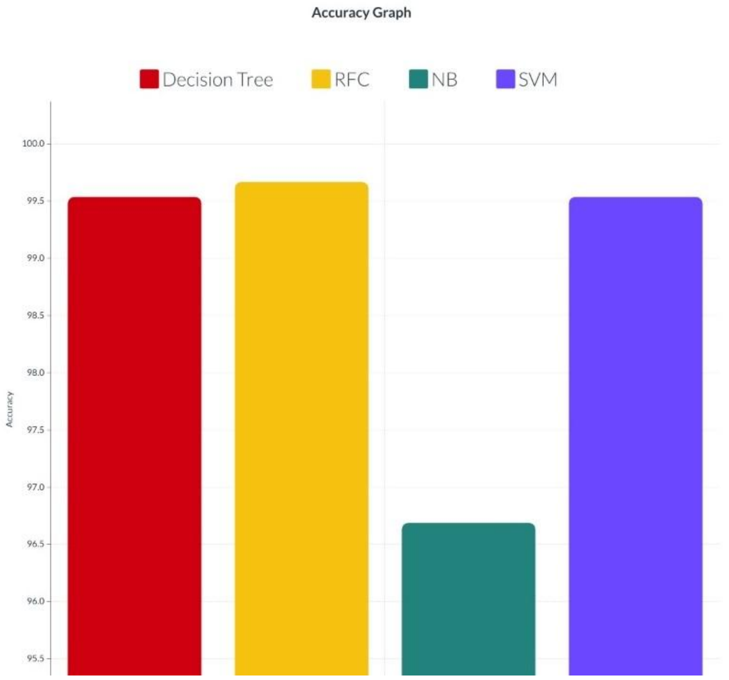
VI. EXPERIMENTS RESULTS

After applying the proposed algorithms, the results from the experiments were calculated and compared. The crime data was classified, predicted and clustered using mentioned algorithms. The performance matrix for the same is given below.

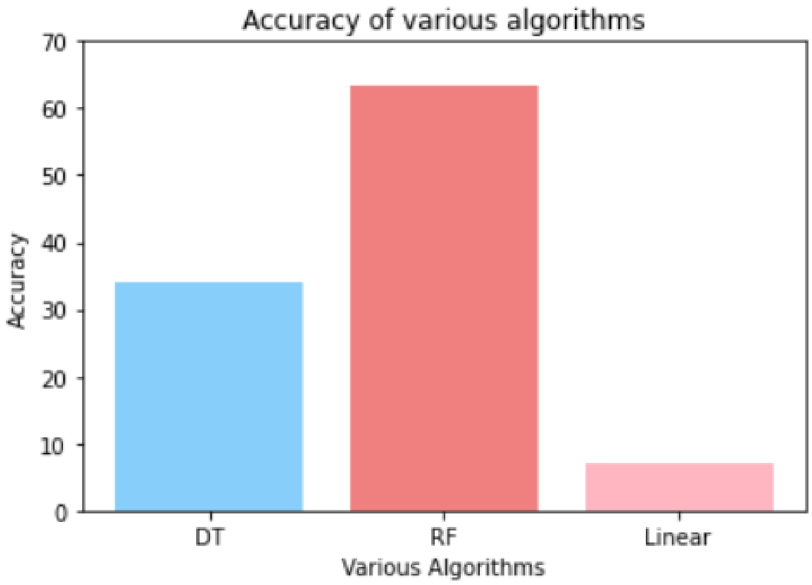
Algorithms	Accuracy	Precision	Recall	F1 Score	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
Classification							
Decision Tree	99.54%	99.53%	99.54%	99.53%	0.49%	0.55%	7.45%
Random Forest	99.67%	99.65%	99.67%	99.65%	0.39%	0.52%	7.23%
Naïve Bayes	96.69%	99.29%	96.69%	97.76%	3.37%	3.50%	18.71%
SVM	99.54%	99.62%	99.64%	99.60%	0.42%	0.55%	7.45%
Regression							
Decision Tree	34.15%	99.46%	99.47%	99.46%			
Random Forest	63.32%						
Multiple Regression	6.86%						

Accuracy Graph for Classification



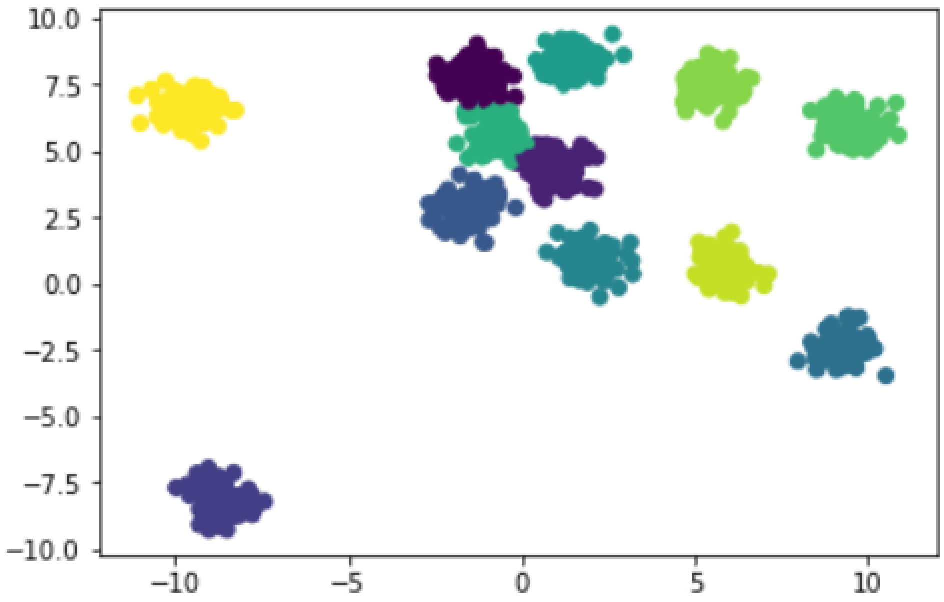


Accuracy Graph for Prediction



These results show that Random Forest is the best classifier and predictor.

BIRCH Algorithm



VII. COMPARATIVE STUDY / RESULTS AND DISCUSSION

In this section the results of the proposed methodology are compared with the existing data mining scheme with Yerpude *et al.* [16]. Yerpude performed predictive analysis and found Random Forest at accuracy 83.39% to be the best algorithm. Our results also show Random

Forest as the best algorithm but with an accuracy of 99.67%. Further we conducted an internal comparison between the algorithms we used to classify, predict and cluster namely Decision Trees, Naive Bayes, Random Forest, Support Vector Machine, Linear Regression and BRICH.

VIII. CONCLUSION AND FUTURE WORK

The proposed work concludes with Random Forest Classifier giving the most balanced results with respect to accuracy, precision, recall and F1 score out of four models for classification, while Decision Tree gave the lowest values in these performance measures. For prediction Random Forest gave the best result, whereas Linear Regression gave the lowest result.

In future, many other data mining algorithms can be applied to understand the data better and provide better solutions to crime. Algorithms like Logistic Regression, K-Nearsest Neighbours, Stochastic Gradient Descent, Hierrachical Clustering can be implemented.

IX. REFERENCES

1. Das, Priyanka & Das, Asit. (2019). *Application of Classification Techniques for Prediction and Analysis of Crime in India.* 10.1007/978-981-10-8055-5_18.
2. Catlett, C., Cesario, E., Talia, D., & Vinci, A. (2019). *Spatio-temporal crime predictions in smart cities: A data-driven approach and experiments.* Pervasive and Mobile Computing, 53, 62-74.
3. David. H, Benjamin & Suruliandi, A. & Raja, S.P.. (2019). *Preventing crimes ahead of time by predicting crime propensity in released prisoners using data mining techniques.* International Journal of Applied Decision Sciences. 12. 307–336. 10.1504/IJADS.2019.100433.
4. Yamini, M. P. C. (2019). *A violent crime analysis using fuzzy c-means clustering approach.* ICTACT Journal on Soft Computing, 9(3), 1939-1944.
5. Sangani, A., Sampat, C., & Pinjarkar, V. (2019, April). *Crime prediction and analysis.* In 2nd International Conference on Advances in Science & Technology (ICAST).
6. Bodare, S., et al. *"Crime Analysis using Data Mining and Data Analytics."* (2019).
7. Das, Priyanka, et al. *"A graph based clustering approach for relation extraction from crime data."* IEEE Access 7 (2019): 101269-101282.

8. Delima, Allemar Jhone P. *"Applying data mining techniques in predicting index and non-index crimes."* International Journal of Machine Learning and Computing 9.4 (2019): 533-538.
9. Nuraeni, Fitri, et al. *"Implementation of K-Means Algorithm with Distance of Euclidean Proximity in Clustering Cases of Violence Against Women and Children."* 2019 1st International Conference on Cybernetics and Intelligent System (ICORIS). Vol. 1. IEEE, 2019.
10. Lavanya, S., and D. Akila. *"PREDICTING CRIMES AGAINST WOMEN'S AND CRIMINAL PERFORMANCE IN TAMILNADU STATE USING CLUSTERING AND CLASSIFICATION ALGORITHM."* Journal of Critical Reviews 7.3 (2020): 548-553.
11. Zhang, Z., Huang, J., Hao, J. et al. *Extracting relations of crime rates through fuzzy association rules mining.* Appl Intell 50, 448–467 (2020.
12. Binnar, Kanchan K., Santosh Kumar, and Nashik SITRC. *"Analysis of The Plaint & FIR Support System using Data Mining."* (Sept 2019 - March 2020)
13. BABU, SRITHA ZITH DEY, DIGVIJAY PANDEY, and ISMAIL SHEIK. *"An overview of a crime detection system using the art of data mining."* International Journal of Innovations in Engineering Research and Technology 7.05 (2020): 125-128.
14. Das, P., Das, A.K., Nayak, J. et al. *A framework for crime data analysis using relationship among named entities.* Neural Comput & Applic 32, 7671–7689 (2020).
15. Singh, Rishabh & Reddy, Rishabh & Kapoor, Vidhi & Churi, Prathamesh. (2020). *K-means Clustering Analysis of Crimes on Indian Women.* 4. 5-25. 10.5281/zenodo.3909955.
16. Yerpude, Prajakta. "*Predictive Modelling of Crime Data Set Using Data Mining."* International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol 7 (2020).
17. Rony, Md, Sagor Chandra Bakchy, and Hadisur Rahman. *"Crime Detection Using Data Mining Techniques."* Computer Science & Engineering: An International Journal (CSEIJ) 10.5 (2020).
18. Vijayarani, S., E. Suganya, and C. Navya. *"Crime analysis and prediction using enhanced Arima model."* Journal homepage: www. ijrpr. com ISSN 2582 (2021): 7421.
19. Kalpana, C. H., and A. Sobhana Rhosaline. *"CRIME RATE PREDICTION BASED ON K-MEANS ALGORITHM."* International Journal of Information Technology (IJIT) 7.2 (2021).

20. Saeed, Shakeel, Muhammad Majid Mahmood Bagram, and Muhammad Munwar Iqbal. *"An Intelligent Analysis of Crime Data using Data Mining Algorithms."* Technical Journal 26.01 (2021): 102-115.
21. Prathap, B. R., Krishna, A. V., & Balachandran, K. (2021). *Crime Analysis and Forecasting on Spatio Temporal News Feed Data—An Indian Context.* In Artificial Intelligence and Blockchain for Future Cybersecurity Applications (pp. 307-327). Springer, Cham.
22. Rastogi¹, I., Jha, A., & Shankar, K. P. *KNOWLEDGE DISCOVERY IN DATABASES FOR PREDICTION OF FUTURE CRIMES.* Turkish Journal of Physiotherapy and Rehabilitation, 32,3.
23. Lavanya, S. "*Determination on Apriori and Clustering Algorithms based on Crime Against Female Permanency-Prediction in Tamil Nadu State."* Turkish Journal of Computer and Mathematics Education (TURCOMAT) 12.10 (2021): 1087-1093.

24. Okeke Ogochukwu, C., and O. Oranyelu Forster. *"AN OVERVIEW OF CRIME ANALYSIS, PREVENTION AND PREDICTION USING DATA MINING BASED ON REAL TIME AND LOCATION DATA."* Published Online February 2021 in IJEAST

25. Kelling, C., Graif, C., Korkmaz, G. et al. *Modeling the Social and Spatial Proximity of Crime: Domestic and Sexual Violence Across Neighborhoods*. J Quant Criminol 37, 481–516 (2021).
26. Rai, M. Tech Scholar Meenu, and Bhawana Pillai. *"Criminal Activities Predictive Analysis Using Data Mining Techniques." (2021).*

27. Panchal, Riddhi, and Binod Kumar. *"Threat Analysis Using Data Mining Technique." Artificial Intelligence and Data Mining Approaches in Security Frameworks* (2021): 197-207.

28. Zahran, Samah, Eman M. Mohamed, and Hamdy M. Mousa. *"Detecting and Predicting Crimes using Data Mining Techniques: Comparative Study."* IJCI. International Journal of Computers and Information 8.2 (2021): 57-62.

29. Hernández, J., Jiménez, D., Zagal, R., Mata, F., & Borges, J. A. L. (2021, November). *Analysis of the Level of Geographic Criminal Risk Oriented to Women. In International Congress of Telematics and Computing (pp. 244-255).* Springer, Cham.

30. H. Kaur, T. Choudhury, T. P. Singh and M. Shamoon, *"Crime Analysis using Text Mining,"* 2019 International Conference on contemporary Computing and Informatics (IC3I), 2019, pp. 283-288, doi: 10.1109/IC3I46837.2019.9055606.

Appendix

Crime against Women analysis

About Dataset It has state-wise and district level data on the various crimes committed against women between 2001 to 2014. Crimes that included are :

Rape Kidnapping and Abduction Dowry Deaths Assault on women with intent to outrage her modesty Insult to modesty of Women Cruelty by Husband or his Relatives Importation of Girl

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
import warnings
warnings.filterwarnings('ignore')
```

Basic information of Data like Shape, data type, null values, Unique Characters

```
data = pd.read_csv(r"C:\Users\LAASYA\Desktop\complete data.csv")
data.shape
```

```
(10186, 13)
```

```
data.head()

data.info()

data.isnull().sum()

data.nunique()

data = data.drop("S.No",axis=1)

data.head()

list(data["STATE/UT"].unique())
```

Data Cleaning

Data Cleaning on STATE/UT Column

In India we have only 36 States/UT but as per the data it has 72 States/UT, so we deep dive into find the errors and fix it, When we see the STATE/UT list we came to know that some of them are entered in upper Case and some in lower case. so the count has increased than actual. To clear this issue we converted all to a lower case and then also it shows 3 No. higher than the actual and while inspecting found that there is spacing issues and then fixed it to obtain the actual.

So, it is always good to compare it with the real data.

```
list(data["STATE/UT"].unique())
len(list(data["STATE/UT"].unique()))
```

```
data.replace({'A&N Islands':'A & N Islands','D&N Haveli':'D & N HAVELI','Delhi UT':'DELHI'},inplace=True)

len(list(data["STATE/UT"].unique()))

data["STATE/UT"]=data["STATE/UT"].str.casefold()

len(list(data["STATE/UT"].unique()))

data["DISTRICT"].value_counts()
```

Data Cleaning on DISTRICT

Initially we converted all the entries to lower case, and then obtained the unique list of District, while inspecting found that it has entries like “total”, “zz total”, “total district(s)”, “delhi ut total” this doesn’t look like an district name so while investigating it seems to be an total value of that state on each crime. If it present in the data it will mess up the data by increasing the No. of cases in each category so, found the indexes of that total and removed from the dataset.

```
data["DISTRICT"]=data["DISTRICT"].str.lower()

data=data.drop(list(data[data["DISTRICT"]=="total"].index))

len(list(data[data["DISTRICT"]=="zz total"].index))

len(list(data[data["DISTRICT"]=="total district(s)"].index))

len(list(data[data["DISTRICT"]=="delhi ut total"].index))

data = data.drop(list(data[data["DISTRICT"]=="zz total"].index))

data = data.drop(list(data[data["DISTRICT"]=="total district(s)"].index))

data = data.drop(list(data[data["DISTRICT"]=="delhi ut total"].index))

data["DISTRICT"].value_counts()

data["DISTRICT"].unique()

data.columns
```

```
Index(['STATE/UT', 'DISTRICT', 'Year', 'Rape', 'Kidnapping and Abduction',
       'Dowry Deaths', 'Assault on women with intent to outrage her modesty',
       'Insult to modesty of Women', 'Cruelty by Husband or his Relatives',
       'Importation of Girls', 'Target', 'Target_Label'],
      dtype='object')
```

```
data["Total"]=data["Rape"]*7 + data["Kidnapping and Abduction"]*6 + data["Dowry Deaths"]*5 +data["Assault on women with intent to outrage her modesty"]*4 +data["Insult to modesty of Women"]*3 + data["Cruelty by Husband or his Relatives"]*2 + data["Importation of Girls"]*1

data.head()
```

Data Analysis

Heat map

```
fig, ax = plt.subplots(figsize=(10,6))
sns.heatmap(data.corr(), center = 0, cmap = "Reds")
ax.set_title("Crime against women Data")
```

```
Text(0.5, 1.0, 'Crime against women Data')
```

```
fig, ax = plt.subplots(figsize=(10,6))
sns.heatmap(data.corr(), center = 0, cmap = "BrBG", annot = True)
```

Year Wise Analysis of Crime

The Below graph depicts the Year wise analysis of crime from 2001 to 2014. It seems like the crime rate increases rapidly as the year goes On.

```
plt.figure(figsize=(12,8))
data.groupby("Year")["Total"].sum().plot.bar()
```

Percentage Distribution of each Crime based on Year

```
data.groupby("Year")["Rape"].sum().plot.pie(autopct='%1.0f%%')
plt.tight_layout()
```

```
data.groupby("Year")["Kidnapping and Abduction"].sum().plot.pie(autopct='%1.0f%%')
plt.tight_layout()
```

```
data.groupby("Year")["Dowry Deaths"].sum().plot.pie(autopct='%1.0f%%')
plt.tight_layout()
```

```
data.groupby("Year")["Assault on women with intent to outrage her
modesty"].sum().plot.pie(autopct='%1.0f%%')
plt.tight_layout()
```

```
data.groupby("Year")["Insult to modesty of Women"].sum().plot.pie(autopct='%1.0f%%')
plt.tight_layout()
```

```
data.groupby("Year")["Cruelty by Husband or his Relatives"].sum().plot.pie(autopct='%1.0f%%')
plt.tight_layout()
```

```
data.groupby("Year")["Importation of Girls"].sum().plot.pie(autopct='%1.0f%%')
plt.tight_layout()
```

Summary

The Below graph represents the year by year trend on each category of Crime.

```
plt.figure(figsize=(20,12))
plt.subplot(2,4,1)
data.groupby("Year")["Rape"].sum().plot(title="Rape")
plt.subplot(2,4,2)
data.groupby("Year")["Kidnapping and Abduction"].sum().plot(title="Kidnapping and Abduction")
plt.subplot(2,4,3)
data.groupby("Year")["Dowry Deaths"].sum().plot(title="Dowry Deaths")
plt.subplot(2,4,6)
data.groupby("Year")["Assault on women with intent to outrage her
modesty"].sum().plot(title="Assault on women with intent to outrage her modesty")
plt.subplot(2,4,5)
```

```
data.groupby("Year")["Insult to modesty of Women"].sum().plot(title="Insult to modesty of
Women")
plt.subplot(2,4,4)
data.groupby("Year")["Cruelty by Husband or his Relatives"].sum().plot(title="Cruelty by Husband
or his Relatives")
plt.subplot(2,4,7)
data.groupby("Year")["Importation of Girls"].sum().plot(title="Importation of Girls")
plt.subplot(2,4,8)
data.groupby("Year")["Total"].sum().plot(title="Total No. of Crimes")
```

```
plt.figure(figsize=(15,7))
data.groupby("Year")["Rape"].sum().plot()
data.groupby("Year")["Kidnapping and Abduction"].sum().plot()
data.groupby("Year")["Dowry Deaths"].sum().plot(label="Dowry Deaths")
data.groupby("Year")["Assault on women with intent to outrage her modesty"].sum().plot()
data.groupby("Year")["Insult to modesty of Women"].sum().plot()
data.groupby("Year")["Cruelty by Husband or his Relatives"].sum().plot()
data.groupby("Year")["Importation of Girls"].sum().plot()
plt.legend()
plt.tight_layout()
```

Yearwise Crime Rate on Different Categories

```
crimes=['Rape','Kidnapping and Abduction','Dowry Deaths',
        'Assault on women with intent to outrage her modesty',
        'Insult to modesty of Women','Cruelty by Husband or his Relatives',
        'Importation of Girls']

data1=pd.DataFrame()
for i in crimes:
    data_crimes=data.groupby(['Year'])[i].sum()
    data1[i]=data_crimes
```

Percentage Contribution of Each Category of Crime

```
a=[]
for i in crimes:
  a.append(data1[i].sum())
a.sort()
plt.figure(figsize=(10,15))
plt.pie(a,labels=crimes,autopct='%1.2f%%',colors=['black', 'gold', 'lightskyblue',
'lightcoral','lightpink','lightcyan','lightgreen'])
plt.tight_layout()

data2 = data1.T
```

Importation of Girls

State/UT wise Analysis of Crime

```
plt.figure(figsize=(17,7))
data.groupby("STATE/UT")["Total"].sum().sort_values(ascending=False).plot.bar()

crimes=['Rape','Kidnapping and Abduction','Dowry Deaths',
        'Assault on women with intent to outrage her modesty',
        'Insult to modesty of Women','Cruelty by Husband or his Relatives',
        'Importation of Girls']

data_state=pd.DataFrame()
```

```
for i in crimes:
    data_state_crimes=data.groupby(['STATE/UT'])[i].sum()
    data_state[i]=data_state_crimes
data_state["Total"]=data_state.sum(axis=1)
data_state = data_state.sort_values(by="Total",ascending=False)
data_state.reset_index()
data_state
```

Top 3 States with Higher Number of Crimes

```
data_state.head(3)
```

Label Encoding and Splitting data into Training and Testing data

Label Encoding

```
inputs = data.drop('Target_Label', axis = 'columns')
inputs= inputs.drop('Target', axis = 'columns')
target = data['Target_Label']

from sklearn.preprocessing import LabelEncoder

le_state = LabelEncoder()
le_district = LabelEncoder()
le_year = LabelEncoder()
le_targetlabel = LabelEncoder()

inputs['STATE/UT_n'] = le_state.fit_transform(inputs['STATE/UT'])
inputs['DISTRICT_n'] = le_district.fit_transform(inputs['DISTRICT'])
inputs['Year_n'] = le_year.fit_transform(inputs['Year'])
target = le_targetlabel.fit_transform(target)

inputs.head()

inputs_n = inputs.drop(['STATE/UT', 'DISTRICT' , 'Year', 'Total'], axis = 'columns')

inputs_n.head()
```

Splitting

```
from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(inputs_n, target ,random_state= 101, stratify=
target, train_size = 0.7)
X_train.shape , X_test.shape
```

Classification

Decision Tree

```
from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import confusion_matrix

dt_model=DecisionTreeClassifier(random_state=10)

dt_model.fit(X_train,Y_train)

DecisionTreeClassifier(random_state=10)

dt_model.score(X_train,Y_train)
dt_model.score(X_test,Y_test)

dt_accuracy = dt_model.score(X_test,Y_test)

accuracy.append(dt_accuracy*100)

result=dt_model.predict(X_test)

data_dt_cls=pd.DataFrame({'Actual':Y_test, 'Predicted':result})
dt_model.predict_proba(X_test)
```

Confusion Matrix

```
from sklearn import metrics

print(confusion_matrix(Y_test, result))

mat = confusion_matrix(result, Y_test)
names = np.unique(result)
sns.heatmap(mat, square=True, annot=True, fmt='d', cbar=False,
            xticklabels=names, yticklabels=names)
plt.xlabel('Truth')
plt.ylabel('Predicted')
```

Precision, Recall and F1 Score

```
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import plot_precision_recall_curve
from sklearn.metrics import precision_score,f1_score
from sklearn.metrics import recall_score

precision = precision_score(Y_test, result,average='weighted')
recall = recall_score(Y_test, result,average='weighted')
score = f1_score(Y_test, result, average='weighted')


print('Precision: ',precision)
print('Recall: ',recall)
print('F1_Score: ',score)
```

Mean Absolute Error, MSE, RMSE

```
print('Mean Absolute Error:', metrics.mean_absolute_error(Y_test, result))
print('Mean Squared Error:', metrics.mean_squared_error(Y_test, result))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(Y_test, result)))
```

Plot a tree

```
from sklearn import tree
decision_tree=tree.export_graphviz(dt_model,out_file='tree.dot',feature_names=X_train.columns,max_depth=10,filled=True)

 dt_model.predict([[1,2,3,4,5,6,7,9,8,0]])
```

Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier

classifier_rf = RandomForestClassifier(n_jobs=-1, max_depth=5,
                                       n_estimators=100, oob_score=True)

classifier_rf.fit(X_train, Y_train)
```

```
RandomForestClassifier(max_depth=5, n_jobs=-1, oob_score=True)
```

```
classifier_rf.oob_score_
```

```
rf = RandomForestClassifier(n_jobs=-1)
```

```
from sklearn.model_selection import GridSearchCV
```

```
params = {
    'max_depth': [2,3,5,10,20],
    'min_samples_leaf': [5,10,20,50,100,200],
    'n_estimators': [10,25,30,50,100,200]
}
```

```
grid_search = GridSearchCV(estimator=rf,
param_grid=params,
cv = 4,
n_jobs=-1, verbose=1, scoring="accuracy")
```

```
grid_search.fit(X_train,Y_train)
```

```
Fitting 4 folds for each of 180 candidates, totalling 720 fits
```

```
GridSearchCV(cv=4, estimator=RandomForestClassifier(n_jobs=-1), n_jobs=-1,
             param_grid={'max_depth': [2, 3, 5, 10, 20],
                         'min_samples_leaf': [5, 10, 20, 50, 100, 200],
                         'n_estimators': [10, 25, 30, 50, 100, 200]},
             scoring='accuracy', verbose=1)
```

```
grid_search.best_score_
```

```
rf_best = grid_search.best_estimator_
rf_best
```

```
RandomForestClassifier(max_depth=5, min_samples_leaf=5, n_jobs=-1)
```

```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[5], feature_names = inputs_n.columns,class_names=['LOW', 'HIGH',
'MEDIUM'],filled=True);
```

```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[5], feature_names = X_train.columns,class_names=['LOW', 'HIGH',
'MEDIUM'],filled=True);
```

```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[7], feature_names = X_test.columns,class_names=['LOW', 'HIGH',
'MEDIUM'],filled=True);
```

```
rf_best.feature_importances_
```

```
imp_df = pd.DataFrame({
    "Varname": X_train.columns,
    "Imp": rf_best.feature_importances_
})
```

```
imp_df.sort_values(by="Imp", ascending=False)

%matplotlib inline

inputs_n.head()

rf.fit(X_train,Y_train)

RandomForestClassifier(n_jobs=-1)

rf.score(X_test, Y_test)

rf_accuracy = rf.score(X_test, Y_test)

accuracy.append(rf_accuracy*100)

result = rf.predict(X_test)

data_rf_cls = pd.DataFrame({'Actual': Y_test, 'Predicted': result})
rf.predict_proba(X_test)
```

Confusion Matrix

```
print(confusion_matrix(Y_test, result))

mat = confusion_matrix(result, Y_test)
names = np.unique(result)
sns.heatmap(mat, square=True, annot=True, fmt='d', cbar=False,
            xticklabels=names, yticklabels=names)
plt.xlabel('Truth')
plt.ylabel('Predicted')
```

Precision, Recall and F1 Score

```
precision = precision_score(Y_test, result,average='weighted')
recall = recall_score(Y_test, result,average='weighted')
score = f1_score(Y_test, result, average='weighted')

print('Precision: ',precision)
print('Recall: ',recall)
print('F1_Score: ',score)
```

Mean Absolute Error, MSE, RMSE

```
print('Mean Absolute Error:', metrics.mean_absolute_error(Y_test, result))
print('Mean Squared Error:', metrics.mean_squared_error(Y_test, result))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(Y_test, result)))

rf.predict([[1,2,3,4,5,6,7,9,8,0]])
```

Naive Bayes

```
from sklearn.naive_bayes import GaussianNB
nb_model = GaussianNB()

nb_model.fit(X_train, Y_train)

GaussianNB()

nb_model.score(X_train,Y_train)
nb_model.score(X_test,Y_test)

nb_accuracy = nb_model.score(X_test,Y_test)

accuracy.append(nb_accuracy*100)

result=nb_model.predict(X_test)

data_nb_cls=pd.DataFrame({'Actual':Y_test, 'Predicted':result})
nb_model.predict_proba(X_test)
```

Confusion Matrix

```
print(confusion_matrix(Y_test, result))

mat = confusion_matrix(result, Y_test)
names = np.unique(result)
sns.heatmap(mat, square=True, annot=True, fmt='d', cbar=False,
            xticklabels=names, yticklabels=names)
plt.xlabel('Truth')
plt.ylabel('Predicted')
```

Precision, Recall and F1 Score

```
precision = precision_score(Y_test, result,average='weighted')
recall = recall_score(Y_test, result,average='weighted')
score = f1_score(Y_test, result, average='weighted')

print('Precision: ',precision)
print('Recall: ',recall)
print('F1_Score: ',score)
```

Mean Absolute Error, MSE, RMSE

```
print('Mean Absolute Error:', metrics.mean_absolute_error(Y_test, result))
print('Mean Squared Error:', metrics.mean_squared_error(Y_test, result))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(Y_test, result)))
```

Plot a tree

```
nb_model.predict([[1,2,3,4,5,6,7,9,8,0]])
```

Support Vector machine

```
from sklearn.svm import SVC
SVM_model = SVC()

SVM_model.fit(X_train,Y_train)

SVC()

SVM_model.score(X_train,Y_train)

SVM_model.score(X_test,Y_test)

SVM_accuracy = SVM_model.score(X_test,Y_test)

accuracy.append(SVM_accuracy*100)

result=SVM_model.predict(X_test)

data_svm_cls=pd.DataFrame({'Actual':Y_test, 'Predicted':result})
```

Confusion matrix

```
print(confusion_matrix(Y_test, result))

mat = confusion_matrix(result, Y_test)
names = np.unique(result)
sns.heatmap(mat, square=True, annot=True, fmt='d', cbar=False,
            xticklabels=names, yticklabels=names)
plt.xlabel('Truth')
plt.ylabel('Predicted')
```

Precision, Recall and F1 Score

```
precision = precision_score(Y_test, result,average='weighted')
recall = recall_score(Y_test, result,average='weighted')
score = f1_score(Y_test, result, average='weighted')

print('Precision: ',precision)
print('Recall: ',recall)
print('F1_Score: ',score)
```

Mean Absolute Error, MSE, RMSE

```
print('Mean Absolute Error:', metrics.mean_absolute_error(Y_test, result))
print('Mean Squared Error:', metrics.mean_squared_error(Y_test, result))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(Y_test, result)))

SVM_model.predict([[1,2,3,4,5,6,7,9,8,0]])
```

Classification accuracy plot

```
models = ("DT" , "RFC" , "NB" , "SVM")

plt.bar(models , accuracy , width = 0.8, color = ['lightskyblue', 'lightcoral','lightpink','cyan'])
plt.title("Accuracy of various algorithms")

plt.xlabel("Various Algorithms")
plt.ylabel("Accuracy")
plt.ylim(96, 100.0)

plt.show()
```

Regression

```
r_accuracy = []
```

Decision Tree

```
from sklearn.tree import DecisionTreeRegressor

dt_regressor = DecisionTreeRegressor(random_state=0)

dt_regressor.fit(X_train,Y_train)

DecisionTreeRegressor(random_state=0)

dt_regressor.predict(X_test)

dt_regressor.score(X_train,Y_train)

dt_regressor.score(X_test,Y_test)
```

```
dt_accuracy = dt_regressor.score(X_test,Y_test)

r_accuracy.append(dt_accuracy * 100)

result_dt_reg=dt_regressor.predict(X_test)

data_dt_reg=pd.DataFrame({'Actual':Y_test, 'Predicted':result_dt_reg})


precision = precision_score(Y_test, result_dt_reg,average='weighted')
recall = recall_score(Y_test, result_dt_reg,average='weighted')
score = f1_score(Y_test, result_dt_reg, average='weighted')
print('Precision: ',precision)
print('Recall: ',recall)
print('F1_Score: ',score)
```

Plot a tree

```
from sklearn import tree
dt_reg_tree =tree.export_graphviz(dt_regressor,out_file='dt_reg_tree.dot',feature_names=X_train.columns,max_depth=10,filled=True)

dt_regressor.predict([[1,2,3,4,5,6,7,9,8,0]])
```

Random Forest

```
from sklearn.ensemble import RandomForestRegressor

rf_regressor = RandomForestRegressor(random_state=0)

rf_regressor.fit(X_train,Y_train)

RandomForestRegressor(random_state=0)

rf_regressor.predict(X_test)

rf_regressor.score(X_train,Y_train)

rf_regressor.score(X_test,Y_test)

rf_accuracy = rf_regressor.score(X_test,Y_test)

r_accuracy.append(rf_accuracy * 100)

result_rf_reg=rf_regressor.predict(X_test)

data_rf_reg=pd.DataFrame({'Actual':Y_test, 'Predicted':result_rf_reg})
```

Linear Regression

```
from sklearn.linear_model import LinearRegression

l_reg = LinearRegression()

l_reg.fit(X_train, Y_train)

LinearRegression()

l_reg.predict(X_test)
l_reg.score(X_train,Y_train)

l_reg.score(X_test,Y_test)

l_reg_accuracy = l_reg.score(X_test,Y_test)

r_accuracy.append(l_reg_accuracy * 100)

result_ln_reg=l_reg.predict(X_test)

data_ln=pd.DataFrame({'Actual':Y_test, 'Predicted':result_ln_reg})
l_reg.predict([[1,2,3,4,5,6,7,9,8,0]])
```

Regression Accuracy plot

```
r_accuracy

r_models = ("DT", "RF", "Linear")

plt.bar(r_models , r_accuracy , width = 0.8, color = ['lightskyblue', 'lightcoral','lightpink'])
plt.title("Accuracy of various algorithms")

plt.xlabel("Various Algorithms")
plt.ylabel("Accuracy")
plt.ylim(0,70)

plt.show()
```

Clustering

BIRCH

```
from sklearn.datasets import make_blobs
from sklearn.cluster import Birch

model=Birch(branching_factor=50,n_clusters=None, threshold=1.5)

model.fit(inputs_n)
Birch(n_clusters=None, threshold=1.5)

pred = model.predict(inputs_n)

plt.scatter(inputs_n['Rape'],pred,cmap = 'rainbow', alpha = 0.7, edgecolors = 'b')
from sklearn.datasets import make_blobs
from sklearn.cluster import Birch
data, clusters = make_blobs(n_samples = 1000, centers = 12, cluster_std = 0.50, random_state = 0)
data.shape
model = Birch(branching_factor = 50, n_clusters = None, threshold = 1.5)
model.fit(data)
pred = model.predict(data)
plt.scatter(data[:, 0], data[:, 1], c = pred)
```