Comparative Study of different Prediction Algorithms for Crime Analysis

Saksham Goyal[1], Aman Arora[2], Nikhil Pratap Singh[3], Pulkit Goel[4], Bindu Garg[5]
[1,2,3,4,5] *Department of Computer Science and business systems, BVDU Pune*

Abstract **- In this modern era, crime is still one of the most prevalent problems in our country. It can be easily said that crime rate has not come down even after 75 years of independence. But, recently lots of efforts are been taken to reduce the crime rate using latest technologies like machine learning, data analysis etc. Crime analysis is defined as a set of systematic analytics processes providing timely and useful information on crime patterns and trends. Data mining is the procedure which include evaluating large pre-existing databases in order to generate new information. The extraction of new information is predicted using existing datasets. In recent years, crime data from different heterogeneous sources have given immense opportunities to the research community to effectively study crime prediction tasks in actual real data. In this research paper, we are going to study different prediction algorithms for crime analysis and implement them on our dataset. We will also compare these algorithms to find out the most suitable one.**

Index Terms **- Crime analysis, Crime prediction, Decision tree, Linear classification, Regression, Clustering.**

1.INTRODUCTION

It is quite obvious that the rate of crimes is increasing day by day in all societies in the world. Also, criminals are becoming technologically sophisticated in committing crimes. One challenge faced by security agencies around the world is collecting and analyzing large volumes of data in order to evaluate crimes and criminals and gain some valuable insights from them. With the evolution in data mining technology, lots of techniques have been developed to perform crime analysis and prediction. With so many techniques to choose from it is imperative to select the correct approach for your model and use the most suitable algorithm to perform the given task. Here, we will discuss several prediction algorithms based on different approaches and compare their performance with each other. In the end we will get the best algorithm to apply on our city crime prediction model. Following algorithms will be discussed in this paper:

1. Decision tree
2. Naïve bayes classifier (Gaussian NB)
3. K – means clustering
4. Random forest
5. Evolutionary algorithms

This research paper is organized in different sections for easy understanding of the concepts. Related work done by other researchers is discussed in section 2. The crime analysis and prediction prerequisites are discussed in section 3. Our own research done on various prediction algorithms is discussed in section 4. Results are discussed in section 5. Conclusion and Future Scope of our work is mentioned in Section 6. The best algorithm will be evaluated based on various parameter like accuracy, precision and recall.

2. RELATED WORK

Various researchers across the world have performed research in data mining field, including data analysis, prediction algorithms etc. But very few efforts have been made in criminology field. Here we will discuss the work carried out by various researchers across the world in the field of crime analysis.

H. Benjamin and A. Suruliandi [1] carried out the survey of various crime analysis methods. They discussed the various research work of different researchers based on specific techniques and discussed their results and accuracy. They categorized the various methods under following sections: Text, content and NLP based methods, crime patterns and evidence-based methods and spatial and geo-location-based methods. Here, our area of concern is crime patterns and evidence-based methods. Their survey provides the good foundation to start our own research in crime analysis field.

Jyoti Agarwal [2] in her research performed crime analysis using k-means clustering algorithm on large crime data set. She used rapid miner tool for analyzing crime rates and anticipation of crime rates using different data mining techniques. The objective of their work is to predict crime based on spatial distribution and extract various crime patterns from the existing data and detection of crime. Their analysis includes tracking of homicide from one year to next.

Rasoul Kiani [3] in his work used the clustering and classification techniques for crime analysis. They proposed a model where analysis and prediction of crime is done through the optimization of outlier detection parameters, which is performed using genetic algorithm. Their work includes the extraction of crime patterns by analysis based on criminal information available. Crime prediction using spatial distribution of existing data and crime recognition. They have discussed their approach in detail and provides good base for further research.

Satya Devan [4] proposed a method which will display probability of crime occurrence and visualize crime prone areas. Their model focuses on the crime factors of each day rather than focusing on crime occurrences. They used SVM classifiers, Naïve bayes and logistic regression for classification of crime factors and crime patterns of each day. The prediction of crime spots is done with the help of decision tree algorithm which will detect the areas with high crime probability. Their method consists of a pattern identification phase which can identify trends and patterns in crime using the Apriori algorithm.

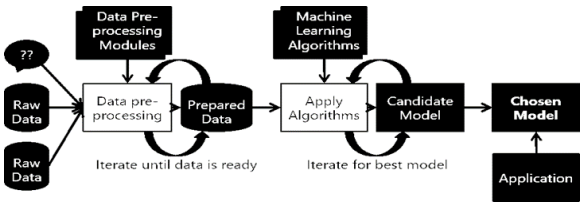
3. PREREQUISITES

3.1. MACHINE LEARNING

Machine Learning refers to the process in which a machine depicts the capability to learn without being explicitly programmed. Machine Learning is an essential skill for any aspiring data scientist who aspire to transform raw data into models of trends and predictions.

Supervised Learning

Supervised learning is that kind of Machine Learning where the model is trained on a labelled dataset. A labelled dataset is one that has both input and output parameters. For example, Decision Tree Algorithm.



Unsupervised Learning

Unsupervised Learning is that kind of Machine Learning where the target to the model is not given while training, i.e., the training model has only the input parameter values. The model has to find which way it can learn by itself. For example, K-Means Algorithm.

3.2. TYPES OF DATA

For Machine Learning, we split the dataset into two, namely training data and testing data.

Training data: It is a part of the actual dataset that is fed to the Machine Learning model to discover and learn patterns. In this way, it trains the model.

Testing Data: Once the Machine Learning model is developed with the help of the training data, a set of unseen data is required to test the model. Here, as the name suggests, the testing data comes into use. It is used to evaluate the performance and progress of the algorithm.

3.3. K-FOLD CROSS VALIDATION

Cross Validation is a process where we resample the dataset to evaluate Machine Learning models. In this process, there is a single parameter, ‘k’ that refers to the number of groups that a given data sample is to be split into. When a specific value of k is chosen, it may be used in place of k in the reference to the model. Like in our study we have divided our dataset into 10 groups, thus performed the 10-fold cross validation.

It is a popular method as it is easy and convenient to understand and as it usually results in a less biased or less optimistic estimate of the model skill than other methods, such as simple train/test split.

The general procedure is as follows:

1. Shuffle the dataset randomly.
2. Split the dataset into k groups.
3. For each unique group:

- Take the group as a hold out or test data set
- Take the remaining groups as a training group as a training data set

- Fit a model on the training set and evaluate it on the test set
- Retain the evaluation score and discard the model

4. Summarize the skill of the model using the sample of model evaluation scores.

Hence, we have used each data of the dataset for the training as well as testing purpose.

3.4 PARAMETERS FOR EVALUATING ALGORITHMS

ACCURACY

Accuracy tells you how many times the ML model was correct overall.

PRECISION

Precision is how good a model is at predicting a specific category

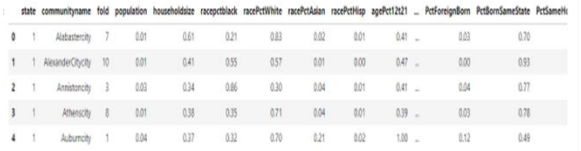
RECALL

Recall tells you how many times a model was able to detect a specific category.

4. METHODOLOGY

4.1. DATASET INFORMATION

Many variables are included in the dataset so that algorithms that select or learn weights for attributes could be tested. However, clearly unrelated attributes were not included. Attributes were picked if there was any plausible connect to crime, plus the attribute to be predicted (like Per Capita Violent crimes). The variables included in the dataset involve the community, such as percent of population considered urban, and the median family income, and also involves law enforcement, such as per capita number of police officers and percent of officers assigned to drug units etc.



	state	communityname	fold	population	householdsize	racepctblack	racePctWhite	racePctAsian	racePctHisp	agePct12t21	...	PctForeignBorn	PctBornSameState	PctSameHo
0	1	Alabastercity	7	0.01	0.61	0.21	0.83	0.02	0.01	0.41	...	0.03	0.70	
1	1	AlexanderCityCity	10	0.01	0.41	0.55	0.57	0.01	0.00	0.47	...	0.00	0.93	
2	1	Annistoncity	3	0.03	0.34	0.86	0.30	0.04	0.01	0.41	...	0.04	0.77	
3	1	Athenscity	8	0.01	0.38	0.35	0.71	0.04	0.01	0.39	...	0.03	0.78	
4	1	Auburncity	1	0.04	0.27	0.32	0.70	0.21	0.02	1.00	...	0.12	0.49	

Data described above is based on original values. All numeric data was normalized into the decimal range 0.00-1.00 using an Unsupervised, equal-interval binning method. Attributes retain their distribution and skew (hence for example the population attribute has a mean value of 0.06 because most communities are small). E.g., An attribute described as 'mean people per household' is actually the normalized (0-1) version of that value.

To begin with, we created a new field ‘highCrime’, which is true if the crime per capita (ViolentCrimesPerPop) is greater than 0.1, and false otherwise. And then we will get the percentage of positive and negative instances in the dataset.

We got the following results:

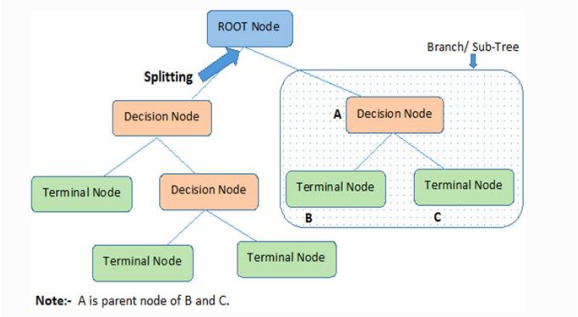
highCrime

False 37.28%

True 62.72%

4.2. DECISION TREE ALGORITHM

The decision tree is a part of supervised learning algorithms. The main purpose for using this algorithm is to create a model that predicts the value of target variable, for which it uses the tree representation to solve the problem, where the leaf node corresponds to a class label and attributes are represented on the internal node of the tree.



STEP 1: We used DecisionTreeClassifier to use decision tree algorithm to predict highCrime on the entire dataset. And then we learned the training accuracy, precision ad recall for this tree.

RESULTS:

Training Accuracy = 1.0 Precision = 1.0 Recall = 1.0

As we can see that, scores values show overfitting so we can define max_depth to avoid the complexity of the tree and to reach a point from where there is a decrease in the cross-validation performance.

STEP 2: Now we have again applied DecisionTreeClassifier after defining max_depth up to 10. And see the results.

RESULTS:

Depth: 1 Accuracy: 0.761

Depth: 2 Accuracy: 0.776
Depth: 3 Accuracy: 0.798
Depth: 4 Accuracy: 0.790
Depth: 5 Accuracy: 0.779
Depth: 6 Accuracy: 0.768
Depth: 7 Accuracy: 0.763
Depth: 8 Accuracy: 0.746
Depth: 9 Accuracy: 0.744
We can see that the point up to which the performance is increasing is the depth 3. We can specify the max_depth = 3 and witness the results

STEP 3: Applying DecisionTreeClassifier with max_depth = 3.

RESULTS:
Accuracy for DT = 0.83592574009
Precision for DT = 0.900260190807
Recall for DT = 0.900260190807

STEP 4: Now apply cross validation score (cross_val_score) to do 10-fold cross-validation to estimate the out-of-training accuracy of decision tree learning for this task.
10-fold cross-validation accuracy, precision and recall

RESULTS:
Cross Validation Accuracy DT: 0.798243718593
Cross Validation Recall DT: 0.843267479959
Cross Validation Precision DT: 0.8392

4.3 NAÏVE BAYES CLASSIFIERS
This is a classification method based on Bayes' theorem, assuming independent predictors. Simply put, the Naive Bayes classifier assumes that the existence of certain features of a class is not correlated with the existence of other features. It is a simple classification algorithm but has high functionality. They find use when the dimensionality of the inputs is high.

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

$$P(B) = \sum_{Y} P(B \mid A)P(A)$$

here,

- P(a|b) is the posterior probability of class (b, target) given predictor (a, attributes).
- P(b) is the prior probability of class.
- P(b|a) is the likelihood which is the probability of predictor given class.
- P(a) is the prior probability of predictor.

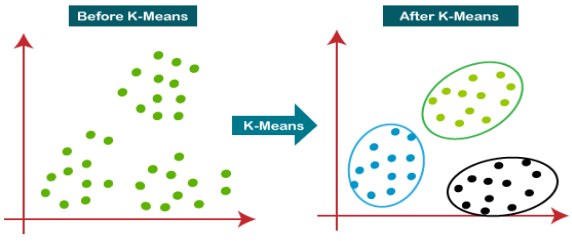
In our research we have used GaussianNB classifier to predict the results. Gaussian Naïve bayes is a variant of naïve bayes that follows Gaussian normal distribution and supports continuous data. Since the data in our dataset all the values are normalized, it is appropriate to use GaussianNB to implement Naïve bayes classifier algorithm.

STEP 1: Use GaussianNB to learn a naïve bayes classifier to predict highCrime. And see the 10-fold cross-validation results of accuracy, precision and recall for this method.

RESULTS:
Accuracy for gaussian NB: 0.761608040201
Recall for gaussian NB: 0.692
Precision for gaussian NB: 0.911799814828

4.4 K-MEANS CLUSTERING ALGORITHM
K-Means clustering is an unsupervised learning algorithm that groups unlabeled data sets into different clusters. where K determines the number of predefined clusters that will be created in the process. For example, if K=2 there are 2 clusters, if K=3 there are 3 clusters, and so on.The algorithm takes an unlabeled data set as input, divides the data set by the number of clusters, and repeats the process until the best cluster is found. In this algorithm, the value of k must be given in advance.

The k-means clustering algorithm basically does two things-
1. Uses an iterative process to determine the best value for the k centroid or centroid
2. Each data point is assigned to the closest k-centroid, and data points near a particular k-centroid creates cluster.



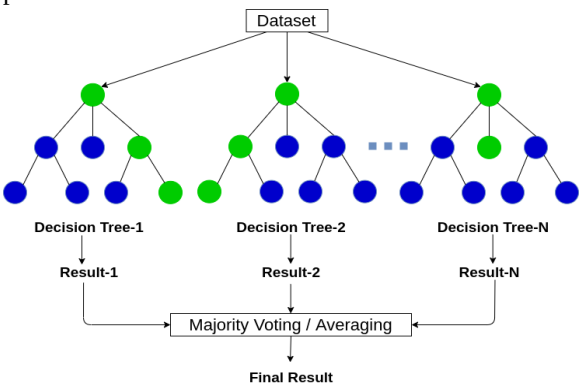
STEP 1: After applying K-means clustering on our dataset, where no. of cluster is set to 2. We calculate the accuracy, precision and recall for this method.

RESULTS:
Accuracy for K-Means: 0.406989949749
Precision for K-Means: 0.612484172735
Recall for K-Means: 0.513012817885

4.5 RANDOM FOREST ALGORITHM

Random Forest is a popular supervised machine learning algorithm for classification and regression problems. I'm building a decision tree for various samples, and in the case of regression I get most of the votes for classification and mean.

One of the most important characteristics of the Random Forest algorithm is that it can handle data sets that contain continuous variables, as in the case of regression, and categorical variables, as in the case of classification. It gives the best results for classification problems.

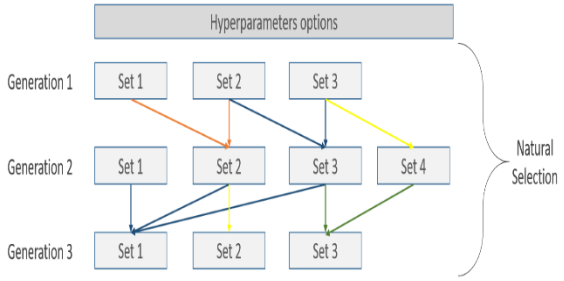


STEP 1: Now we will use RandomForestClassifier to apply random forest algorithm on our dataset and calculate the 10-fold cross-validation accuracy, precision and recall.

RESULTS:
Accuracy for RandomForestClassifier is 0.817963874097
Precision for RandomForestClassifier is 0.843663428685
Recall for RandomForestClassifier is 0.872107936508

4.6 EVOLUTIONARY ALGORITHMS

Evolutionary algorithms are the algorithms inspired by the Darwinian evolution. It solves the problem through processes that emulates the behavior of living organisms. In this algorithm, firstly, the mix of potential solutions to a problem are populated randomly. Next, it is tested for fitness (speed and efficiency) and the fittest individual is selected for reproduction. Then, least fit individuals are eliminated. In the end, we get the fittest individual, i.e., best parameters that gives the most optimum result.



As we have seen above, supervised machine learning techniques are most suitable for crime analysis. Here, we have evaluated three supervised learning algorithms, namely Naïve bayes algorithm, Decision tree algorithm and Random Forest algorithm. Since, Naïve bayes is already giving the best results among three, so we will take random forest algorithm to further optimize it using evolutionary algorithm.

4.7 ENHANCING THE ALGORITHM

There are several variations, but in general, the steps to follow look like this:

- Generate a randomly sampled population (different sets of hyperparameters); this is generation 0.
- Evaluate the fitness value of each individual in the population, in terms of machine learning, get the cross-validation scores.
- Generate a new generation by using several genetic operators. Repeat steps 2 and 3 until a stopping criterion is met.

To enhance the algorithm using this approach, we will create three estimators for random forest classifier, each having different values for 'scoring' parameter. Value of scoring parameter for Estimators 1,2 and 3 will be accuracy, precision and recall respectively.

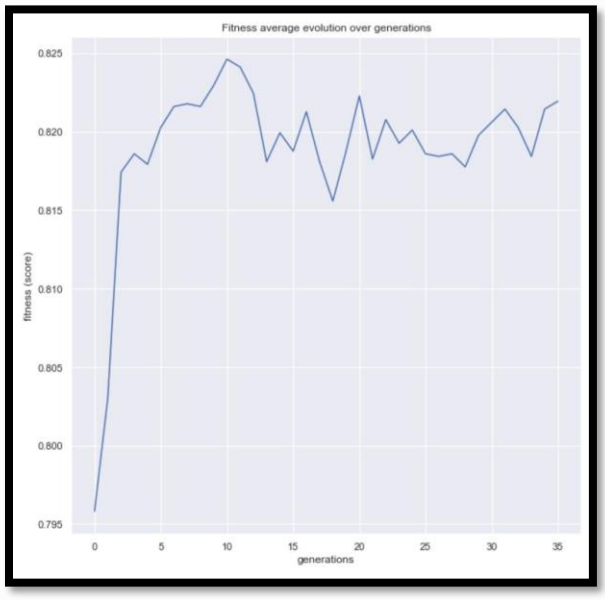
ESTIMATOR 1 – ACCURACY

Generations for Estimator 1

gen	nevals	fitness	fitness_std	fitness_max	fitness_min
0	10	0.829983	0.0134265	0.847571	0.80402
1	20	0.834506	0.0113804	0.847571	0.81407
2	17	0.838191	0.007756	0.847571	0.819095
3	18	0.843216	0.00390684	0.849246	0.837521
4	18	0.846064	0.00321765	0.850921	0.839196
5	19	0.845561	0.0024618	0.850921	0.840871
6	20	0.847404	0.00338756	0.852596	0.842546
7	15	0.848911	0.00324803	0.852596	0.844221
8	18	0.849246	0.00290126	0.852596	0.844221
9	18	0.850921	0.00198194	0.854271	0.847571
10	17	0.849916	0.00352953	0.852596	0.840871
11	18	0.848744	0.00474069	0.852596	0.840871
12	19	0.850586	0.00365451	0.852596	0.842546
13	16	0.847906	0.00564566	0.852596	0.835846
14	18	0.850419	0.00389605	0.854271	0.842546
15	19	0.849581	0.00365451	0.854271	0.844221
16	17	0.851926	0.00282283	0.857621	0.845896
17	20	0.851256	0.0040201	0.857621	0.842546
18	18	0.850586	0.00603015	0.857621	0.835846
19	13	0.855779	0.00158023	0.857621	0.854271
20	19	0.852094	0.00674399	0.857621	0.835846
21	17	0.854774	0.00280788	0.857621	0.847571
22	19	0.854941	0.00282283	0.857621	0.847571
23	17	0.852429	0.00671899	0.857621	0.839196
24	16	0.853099	0.00560826	0.857621	0.844221
25	20	0.855276	0.0049236	0.857621	0.842546
26	18	0.856616	0.0021451	0.857621	0.850921
27	18	0.856114	0.00354935	0.857621	0.845896
28	19	0.849079	0.00724153	0.857621	0.839196
29	19	0.845729	0.00663494	0.857621	0.835846
30	15	0.847571	0.00686563	0.857621	0.840871
31	16	0.851089	0.00659252	0.857621	0.842546
32	17	0.854271	0.00449461	0.857621	0.847571
33	18	0.852429	0.0052142	0.857621	0.845896
34	20	0.850921	0.00555549	0.857621	0.844221
35	16	0.851256	0.005233	0.857621	0.845896

This table represents the value of different generations for estimator 1. We had set up the value of generation parameter = 35, hence we got 35 generations. Now we will plot this result to evaluate how the fitness of the modified estimator increases or decreases over the generations.

Fitness Evaluation Plot for Estimator 1



This plot shows how the fitness increases quickly for the first few generations and then maintains the fitness till the end.

Enhanced Result for Estimator 1

Accuracy for Evolved Estimator 1 = 0.8202005730659025

Precision for Evolved Estimator 1= 0.8464125560538116

Recall for Evolved Estimator 1= 0.8688147295742232

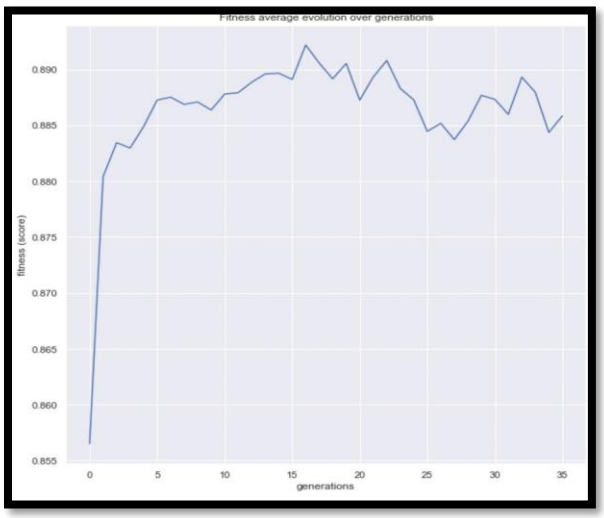
ESTIMATOR 2 – PREDICTION

Generations for Estimator 2

gen	nevals	fitness	fitness_std	fitness_max	fitness_min
0	10	0.85651	0.0225268	0.88763	0.825101
1	16	0.880409	0.0071562	0.890249	0.871972
2	17	0.883436	0.00563017	0.890249	0.874146
3	19	0.882964	0.00607025	0.890249	0.873641
4	19	0.884876	0.00387973	0.890249	0.876489
5	19	0.887252	0.0037173	0.891244	0.877406
6	17	0.887508	0.00547232	0.891244	0.872397
7	18	0.886862	0.00393962	0.891244	0.879049
8	19	0.887076	0.00406762	0.890249	0.879049
9	17	0.886362	0.00552124	0.893747	0.878692
10	20	0.887792	0.00539105	0.893747	0.880922
11	17	0.887912	0.00450755	0.893747	0.883241
12	19	0.888844	0.00459033	0.893747	0.883304
13	19	0.889589	0.00674936	0.893747	0.875236
14	17	0.889656	0.00592146	0.893747	0.873894
15	19	0.889098	0.00841425	0.893747	0.870308
16	18	0.892178	0.00354786	0.893747	0.882199
17	19	0.890562	0.00493293	0.893747	0.881957
18	19	0.889154	0.00588669	0.893747	0.878605
19	18	0.890524	0.00510391	0.893747	0.880457
20	19	0.887247	0.00729659	0.893747	0.87461
21	19	0.88932	0.0059193	0.893747	0.873908
22	19	0.890788	0.00303096	0.893747	0.886692
23	19	0.888295	0.00567353	0.893747	0.876084
24	18	0.887265	0.00727303	0.893747	0.872341
25	19	0.884452	0.00789997	0.893747	0.872295
26	19	0.885172	0.00777023	0.893747	0.875114
27	19	0.883718	0.00672226	0.893747	0.876072
28	17	0.885332	0.00535158	0.894536	0.876386
29	18	0.887674	0.00589221	0.897172	0.88039
30	18	0.887318	0.00550636	0.897172	0.88039
31	17	0.885976	0.00664879	0.897172	0.874919
32	18	0.889299	0.00675202	0.897172	0.880317
33	17	0.887949	0.00495925	0.897172	0.880317
34	19	0.884352	0.00269817	0.886801	0.880317
35	18	0.885831	0.00197093	0.888334	0.880317

This table represents the value of fitness for different generations generated by estimator 2. Here, we have generated only 35 generations and will plot this table to evaluate fitness of this model.

Fitness Evaluation Plot for Estimator 2



Here also, we can see that fitness increases rapidly over first few generations and then maintains it value. The fitness is highest for generation 16.

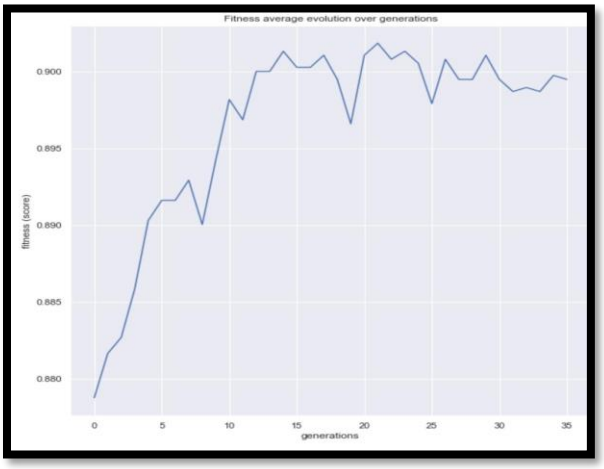
Enhanced Result for Estimator 2
Accuracy for Evolved Estimator 2 = 0.8173352435530086
Precision for Evolved Estimator 2= 0.8449438202247191
Recall for Evolved Estimator 2= 0.8653624856156502

ESTIMATOR 3 – RECALL
Generations for Estimator 3

gen	nevals	fitness	fitness_std	fitness_max	fitness_min
0	10	0.87874	0.0048111	0.884514	0.868766
1	20	0.881627	0.00274024	0.887139	0.87664
2	19	0.882677	0.00455364	0.892388	0.874016
3	16	0.885827	0.00393701	0.892388	0.879265
4	20	0.890289	0.00436043	0.895013	0.88189
5	20	0.891601	0.0023622	0.895013	0.887139
6	18	0.891601	0.00353113	0.895013	0.884514
7	17	0.892913	0.00327821	0.895013	0.884514
8	20	0.890026	0.00414166	0.895013	0.88189
9	18	0.894226	0.00333034	0.897638	0.887139
10	19	0.898163	0.00257164	0.902887	0.895013
11	19	0.89685	0.00470249	0.902887	0.887139
12	19	0.9	0.00341207	0.902887	0.892388
13	20	0.9	0.00397185	0.902887	0.892388
14	17	0.901312	0.00267665	0.902887	0.895013
15	20	0.900262	0.00469516	0.902887	0.889764
16	16	0.900262	0.00352137	0.902887	0.892388
17	19	0.90105	0.00288714	0.902887	0.895013
18	20	0.899475	0.00575635	0.902887	0.887139
19	19	0.896588	0.00553053	0.902887	0.889764
20	16	0.90105	0.00407459	0.902887	0.889764
21	19	0.901837	0.00314961	0.902887	0.892388
22	20	0.900787	0.00419948	0.902887	0.892388
23	19	0.901312	0.00336122	0.902887	0.892388
24	17	0.900525	0.00360833	0.902887	0.895013
25	19	0.8979	0.00603675	0.902887	0.887139
26	19	0.900787	0.00257164	0.902887	0.895013
27	19	0.899475	0.00372111	0.902887	0.895013
28	18	0.899475	0.00353113	0.902887	0.895013
29	19	0.90105	0.00288714	0.902887	0.895013
30	18	0.899475	0.00372111	0.902887	0.892388
31	20	0.898688	0.00457627	0.902887	0.889764
32	18	0.89895	0.00458379	0.902887	0.889764
33	19	0.898688	0.00442318	0.902887	0.892388
34	18	0.899738	0.00282686	0.902887	0.895013
35	20	0.899475	0.00311663	0.902887	0.895013

This table shows the fitness value of the model over the different generations. Here, the value of scoring parameter is recall. Now, we will plot this table on the graph to evaluate the fitness of the model.

Fitness Evaluation Plot for Estimator 3



This plot shows slightly different result compared to above two plots. Here, the fitness is not increased as rapidly as in the above graphs and the values are more fluctuating. But the plot still shows that fitness is increasing as more generations are generated.

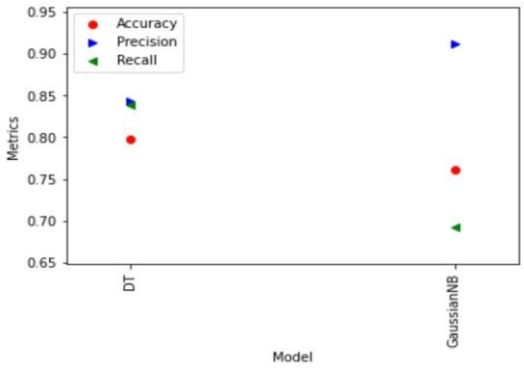
Enhanced Result for Estimator 3
Accuracy for Evolved Estimator 3 = 0.7929799426934098
Precision for Evolved Estimator 3 = 0.8020833333333334
Recall for Evolved Estimator 3 = 0.8860759493670886

5. RESULTS AND DISCUSSIONS

We have applied four different algorithms on our dataset to predict the values and calculated the accuracy, precision and recall for these algorithms. Now, we will plot the results and compare the algorithms with each other and find out the most suitable one to apply on our model.

5.1 DECISION TREE VS GAUSSIAN NAÏVE BAYES

RESULTS:



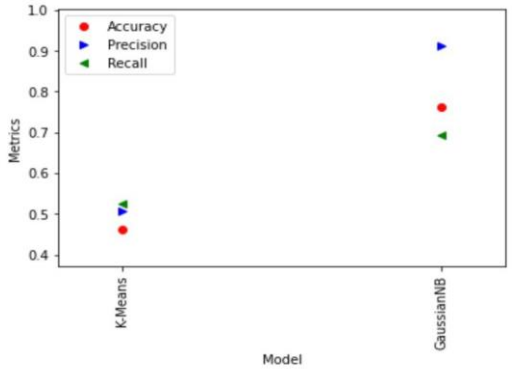
We can see that the accuracy and recall are higher in case of Decision Trees but the precision is higher in case of Gaussian NB.

We have to consider different criteria for choosing a classifier depending upon what we are predicting. Accuracy is not always the most appropriate method to choose the classifier. Here, in case of Crime Prediction, Precision would be the best criteria to compare and it is higher in case of Gaussian NB so the results are better in this case.

So, we will consider Gaussian NB to further compare with other algorithms.

5.2 GAUSSIAN NAÏVE BAYES VS K-MEANS ALGORITHM

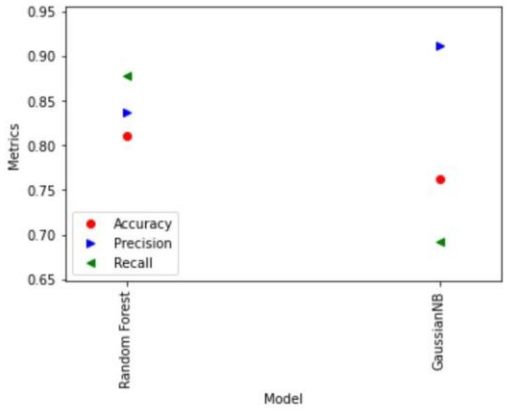
RESULTS:



Here, we can see that Naive Bayes is way better than K-means algorithm, as all the parameters i.e., accuracy, precision and recall are better in case of Gaussian NB. Hence, we are taking Gaussian NB further in our study.

5.3 GAUSSIAN NAÏVE BAYES VS RANDOM FOREST ALGORITHM

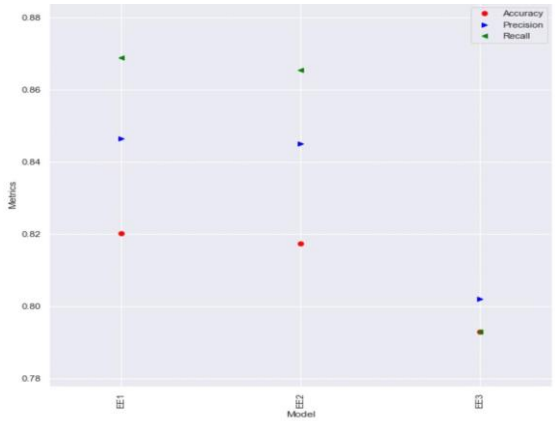
RESULTS:



This case is pretty much similar to the one when we compare Naive bayes to the Decision Tree algorithm where the Accuracy and Recall of the other model is better but Naïve bayes has higher precision. Hence, due to higher precision we can conclude that Naive Bayes algorithm is better than Random Forest algorithm for our prediction model.

5.4 ESTIMATOR 1 VS ESTIMATOR 2 VS ESTIMATOR 3

RESULTS:

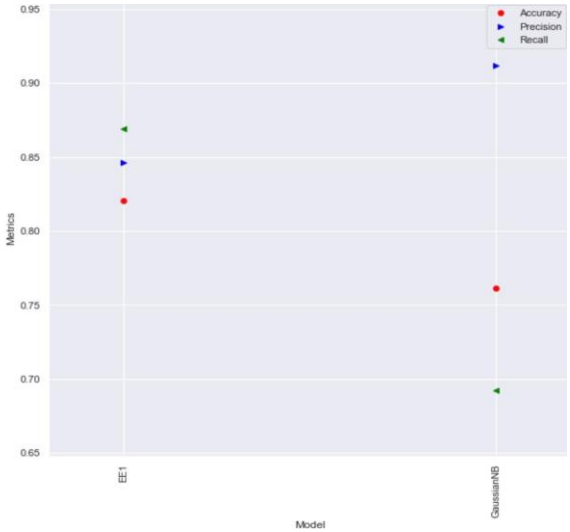


Here we have plotted the accuracy, precision and recall values for all the three estimators. It is very evident that values for Estimator 1 and Estimator 2 are comparable but values for Estimator 3 are much below the required values. So, from this plot it is clear that we should take Estimator 1 to use further in our research as it is showing best values among three.

Now, we will plot the comparison between Estimator 1 and Gaussian Naïve bayes as they shows best results till now.

5.5 ESTIMATOR 1 VS GAUSSIAN NAÏVE BAYES

RESULTS:



Here, we can clearly see that overall values for Estimator 1 are better than any other algorithm used in our research, but the value of precision is still highest in case of gaussian naïve bayes. Since, we are considering precision as our evaluating criteria to choose algorithm for our prediction model we can say that Gaussian naïve bayes slightly overshadow our evolved estimator.

6. CONCLUSION AND FUTURE SCOPE

In this paper, we have studied some known approaches for crime analysis and prediction concerned with data mining. We studied several papers with background in the crime prediction and criminal identification with a theoretical study. And then we carried our own research by comparing several algorithms with each other.

We have seen the results of comparison of different algorithms based on the accuracy, precision and recall values of these algorithms. The algorithms considered were; Decision Tree, Naïve bayes classifier (Gaussian NB), K-means clustering algorithm and Random Forest algorithm. Then we plotted the results of these comparison for clear evaluation of results.

After getting these results, we took an evolutionary algorithm approach to further improve our algorithms. We created three different estimators to improve our algorithm on the basis of three parameters i.e., accuracy, precision and recall. Then we compared these estimators with above algorithms and plotted the results for the same.

After evaluating the results, we concluded that Gaussian Naïve Bayes is the best algorithm among all the algorithms taken into consideration. We have taken precision as a deciding parameter and Gaussian NB proves to be the best in that area.

Now we further plan to develop an "Indian Crime Analytics framework" for law enforcement agencies in India. And this system needs a perfect prediction algorithm to develop a city crime prediction model. This research has already given us the best algorithm i.e., Gaussian Naïve Bayes to use in our prediction model.

REFERENCE

[1] H. Benjamin Fredrick David and A Suruliandi, "Survey on crime analysis and prediction using data mining techniques", ICTAT Journal on Soft Computing.

[2] Jyoti Agarwal, Renuka Nagpal, Rajni Sehgal, "Crime analysis using K-means Clustering", International Journal of computer application.

[3] Rasoul Kiani, Siamak Mahdavi, Ameen Keshavarzi, "Analysis and Prediction of crimes by clustering and Classification", International Journal of Advanced Research in Artificial Intelligence

[4] Shiju Sathyadevan, M.S. Devan and S. Surya Gangadharan, "Crime analysis and prediction using data mining"

[5] Malathi A., Dr. S Santhosh Baboo, "An enhanced Algorithm to predict future crime using data mining", International journal of Computer application

[6] Shanjana A.S and Dr. R. Porkodi, "Crime analysis and prediction using Data Mining: A Review", International Journal of creative research thoughts (IJCRT)

[7] Kaumalee Bogahawatte and Shalinda Adikari. "Intelligent Criminal Identification System", Proceedings of IEEE International Conference on Computer Science and Education, 633-638, 2013

[8] Bogomolov, Andrey and Lepri, Bruno and Staiano, Jacopo and Oliver, Nuria and Pianesi, Fabio and Pentland, Alex.2014. Once upon a crime: Towards crime prediction from demographics and mobile data, Proceedings of the 16th International Conference on Multimodal Interaction.

[9] Yu, Chung-Hsien and Ward, Max W and Morabito, Melissa and Ding, Wei.2011. Crime forecasting using data mining techniques, pages 779-786, IEEE 11th International Conference on Data Mining Workshops (ICDMW)

[10] Kianmehr, Keivan and Alhajj, Reda. 2008. Effectiveness of support vector machine for crime hot-spots prediction, pages 433-458, Applied Artificial Intelligence, volume 22, number 5.

[11] Toole, Jameson L and Eagle, Nathan and Plotkin, Joshua B. 2011 (TIST), volume 2, number 4, pages 38, ACM Transactions on Intelligent Systems and Technology

[12] Wang, Tong and Rudin, Cynthia and Wagner, Daniel and Sevieri, Rich. 2013. pages 515- 530, Machine Learning and Knowledge Discovery in Databases

[13] Friedman, Jerome H." Stochastic gradient boosting." Computational Statistics and Data Analysis 38.4 (2002): 367-378.sts

[14] Leo Breiman, Random Forests, Machine Learning, 2001, Volume 45, Number 1, Page 5.