VISUALIZATION:

The Cyber crime data collected from National Cyber Record Bureau is visualized using the python libraries like matplotlib, seaborn, plotly.

1. The pie chart in figure (1) shows the percentage of each Cybercrime from 2008 to 2019 for all States and Union Territories. From the chart it can be concluded that only Computer related offences are 48.5% of the total Cybercrimes, while other offences under IT Act, 2000 that are not mentioned in the sheets from 2008 to 2019 account for 20.1% of total cybercrime, 17.4% of Cybercrime is of publication or transmission of obscene / sexually explicit material in electronic form and other offences contribute for less than 10%.

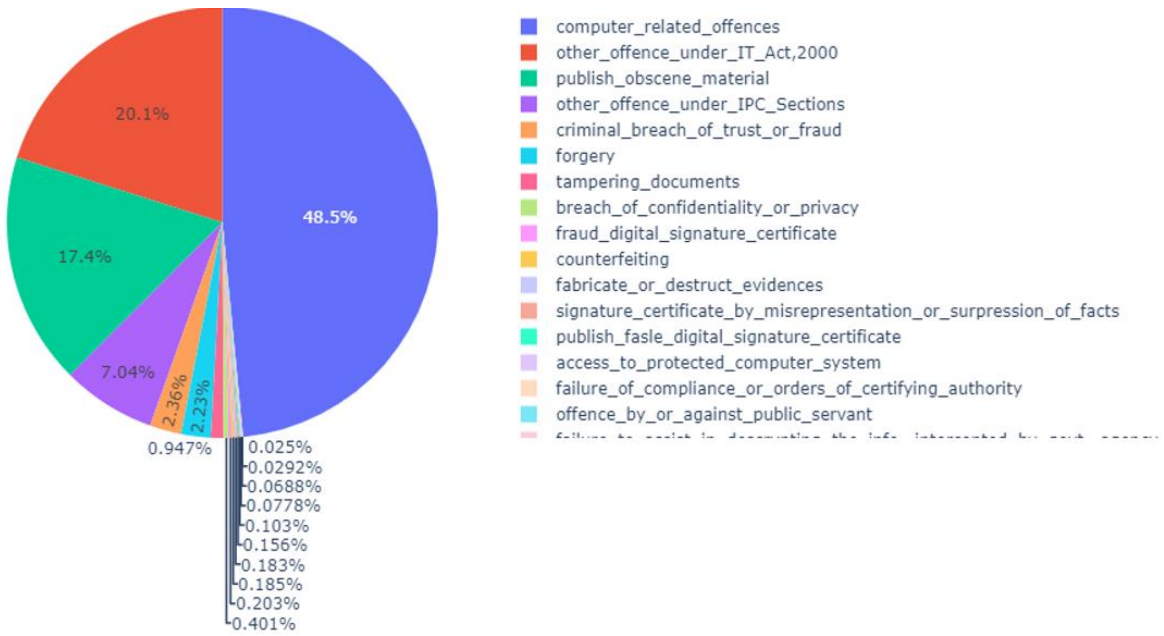


Figure (1)

2. The pie chart in the figure (2) shows the percentage of total Cybercrime in States and Union Territories from 2008 to 2019. From the chart it can be concluded that percentage of Cyber crime is highest in Uttar Pradesh i.e., 23.4% followed by Karnataka 21.9% and Maharashtra 14% while the Cybercrime percentage in other States and UTs are less than 6%.

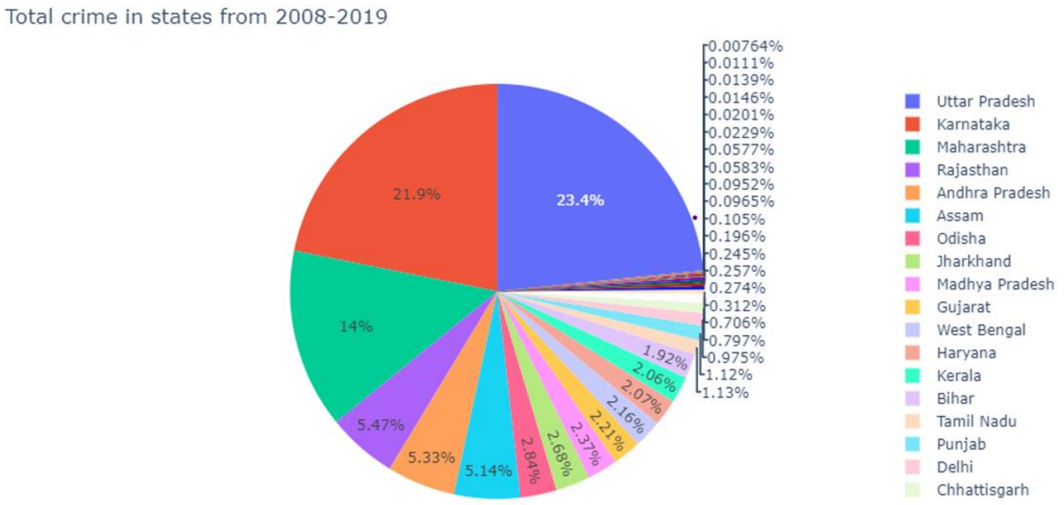
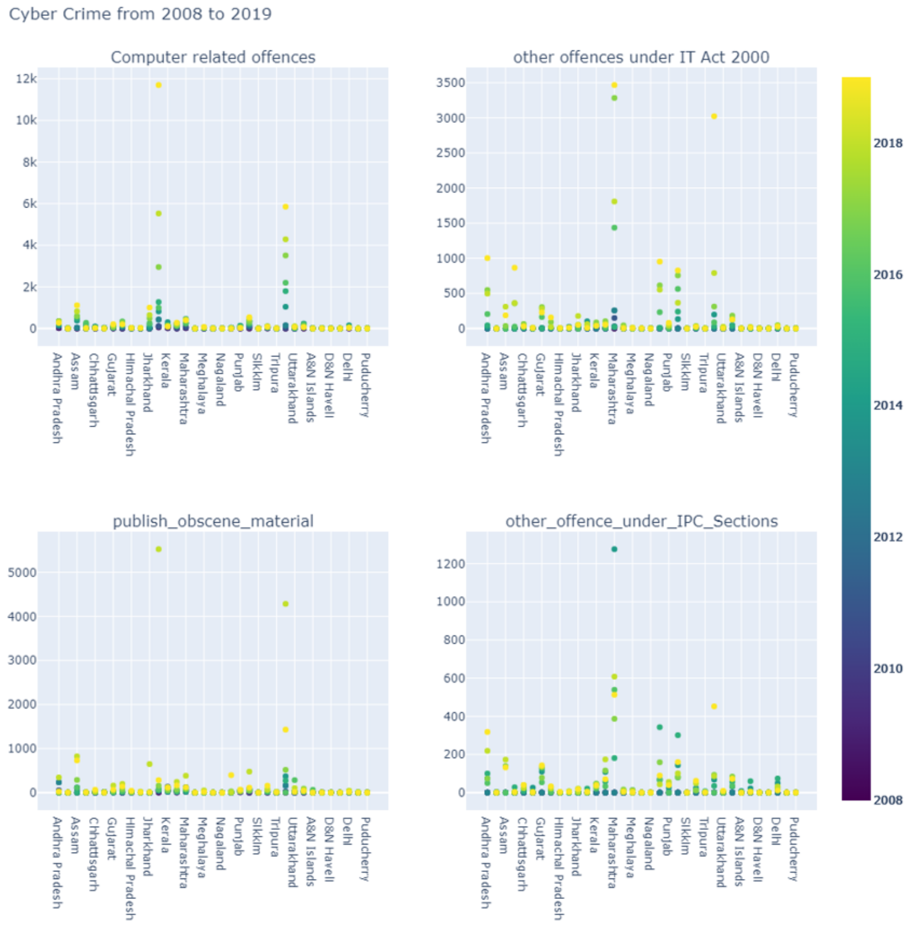


Figure (2)

From figure (1) and figure (2) it is clear that only few crimes contribute for more than 75% of Cyber crimes and same for States and UTs that only few of them are more vulnerable to Cybercrime attacks. So, for the better visualisation of these few crimes one can refer to the figure ().

Figure (3)



The figure (3) above consists of scatter plots of four types of Cyber crimes from 2008 to 2019. It is clearly visible that the graphs of all cyber crimes are increasing with the year passing. Almost all the crimes are highest in Karnataka, Maharashtra and Uttar Pradesh.

@@@

Theory of model used:

Gradient Boosting Regression: Gradient Boosting is one of the machine learning technique used specially for regression and classification problems. It works by combining the predictions from a number of decision trees to get the required prediction. Every new decision tree in this algorithm takes into account the mistakes of the previous tree and try to reduce the error. The nodes in all decision trees are not same they depend upon the least impurity. Different nodes for decision trees help in determining different possible relation among features of dataset.

Maths behind the algorithm:

- Gradient Boosting regression algorithm starts by making a single leaf that is the average of independent variable. For now, the model is predicting the values as the average value of the independent variable. This single leaf is considered as first tree for the of the algorithm.
- Next, error is calculated by subtracting the predicted average value from the actual value and is called the Residual. Using the values of residuals next decision tree is made. The double values in the leaf are replaced by their mean.
- To avoid low bias and high variance problem the model adds some penalty to each decision tree termed as learning rate. One can alter value of learning rate that suits their model.
- In this way, one again can calculate residuals and continue making new decision tree and adding their value to the prediction with some learning rate.
- So, by taking small steps by reducing the error the Gradient Boosting Regression algorithm works.

This paper uses scikit learn library of python to make the prediction. Scikit learn has module ensemble that provides method GradientBoostingRegressor that take cares of all the maths. GradientBoostingRegressor method takes some parameters that can be changed to get optimized prediction results like learning_rate, loss, n_estimators, min_sample_split, etc.

PAPER 1

PAPER TITLE: “Computational System to Classify Cyber Crime offenses using Machine Learning”

AUTHOR: Rupa Ch, Thippa Reddy Gadekallu, Mustufa Haider Abidi and Abdulrahman Al-Ahmari

SUMMARY: The dataset used in this paper is collected from Kaggle and CERT-In 2000 records. Attributes of the dataset are Incident, Offender, Access Violation, Victim, Harm, Year, Location, Age of Offender. The proposed methodology of this paper divides it into 4 steps that are

1. Data Collection: the data is collected from Kaggle and CERT-In 2000 records
2. Preprocessing: This step includes the feature extraction process by using TFIDF vector method and either imputing or removing of null valued columns.

3. Applying model: They have used naïve Bayes algorithm for classification and K-means for clustering. Different cybercrime offences are clustered into some groups.
4. Prediction and Result: For prediction algorithms used are LinearSVC, LogisticRegression, MultinomialNB, RandomForestClassifier and except RandomForestClassifier all other algorithms perform approx. 99% well.

In the future, proposed model can be improved by using deep learning concepts.

PAPER 2

PAPER TITLE: "A Brief Study on Cyber Crimes and IT Act in India"

AUTHOR: Dr. Adv. Mrs. Neeta Deshpande

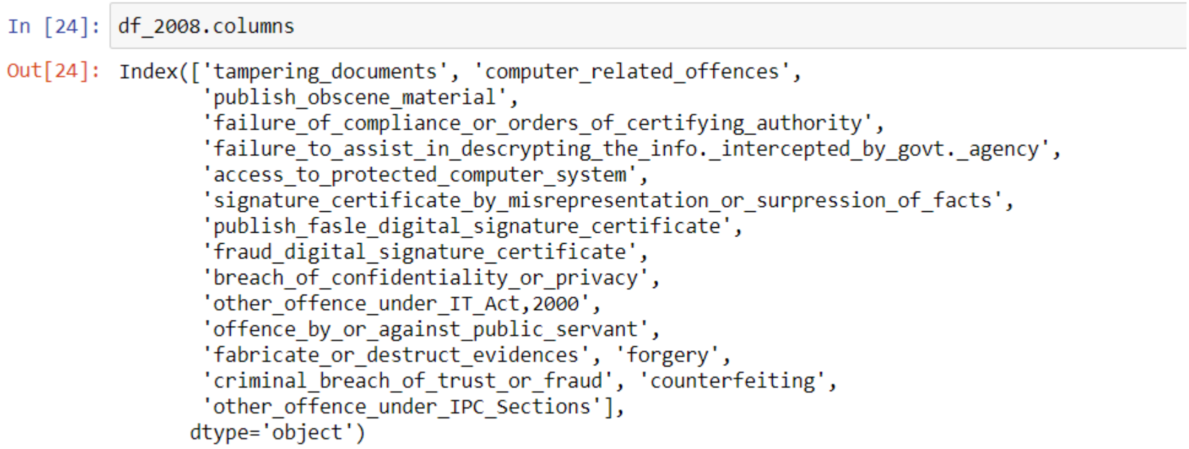
SUMMARY: The paper works with both primary and secondary collected data, primary data is collected by discussions with advocates and secondary data is collected through web sites, e-journals, research papers and other resources. This paper at first visualises number of internet users in the world, in Asia, registered cyber crime cases in India and cases registered and person arrested in States of India. It concludes that the total cyber crime cases registered from 2014 to 2017 are highest in Maharashtra followed by Uttar Pradesh and Karnataka. Then paper discusses about various sections under IT Act 2000 followed by table showing opinions of advocates regarding provisions of IT Act if it is sufficient for tackling all types of cyber crimes or not, etc. Finally, paper concludes by mentioning general and specific suggestions to deal with emerging cybercrimes.

This paper predicts the values for different type of cybercrimes under IT Act, 2000 and IPC sections for year 2020 using the data from 2008 to 2019. This work uses data analysis, data visualization and machine learning techniques to predict data for 2020 of all States and Union Territories. The language used for implementing various regression algorithms is python, regression models Gradient Boosting regression, ridge regression is applied for prediction of cybercrime in 2020. The methodology and results used here can be described in following steps:

Data Collection and Preprocessing → Data Analysis → Using Regression models → Prediction and Analysing Predictions

Data collection and preprocessing:

The data of cybercrime is collected from National Crime Records Bureau (ncrb.gov.in). It contains the record for different types of cybercrimes from 2008 to 2019. The type of cybercrimes used in this work after following preprocessing step to get common columns for all years are:

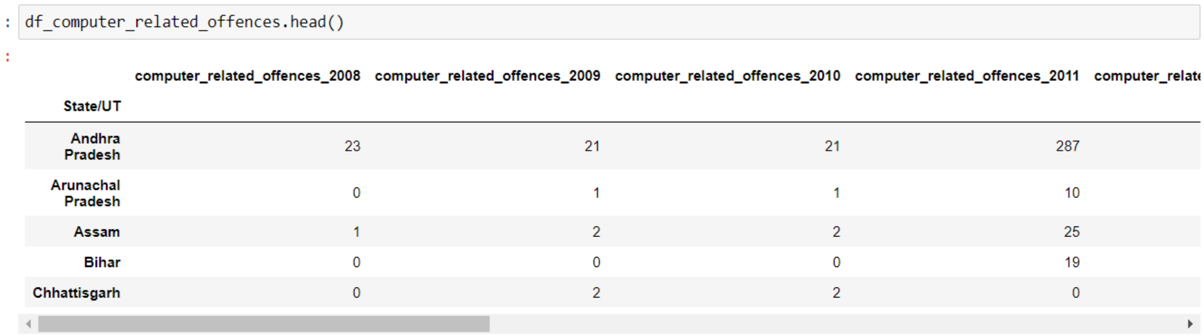


```
In [24]: df_2008.columns

Out[24]: Index(['tampering_documents', 'computer_related_offences',
                'publish_obscene_material',
                'failure_of_compliance_or_orders_of_certifying_authority',
                'failure_to_assist_in_descrypting_the_info._intercepted_by_govt._agency',
                'access_to_protected_computer_system',
                'signature_certificate_by_misrepresentation_or_surpression_of_facts',
                'publish_fasle_digital_signature_certificate',
                'fraud_digital_signature_certificate',
                'breach_of_confidentiality_or_privacy',
                'other_offence_under_IT_Act,2000',
                'offence_by_or_against_public_servant',
                'fabricate_or_destruct_evidences', 'forgery',
                'criminal_breach_of_trust_or_fraud', 'counterfeiting',
                'other_offence_under_IPC_Sections'],
               dtype='object')
```

Data Analysis: (we can write statistics analysis taken using pandas example the states with almost no crime and the once with highest also same for type of crime)

For analysing, dataframes are created for all types of cybercrimes from 2008 to 2019 for all States and Union territories. Like figure () shows the dataframe of cybercrime computer related offences for all years. All these dataframes all also used for prediction of particular type of cybercrime.



```
: df_computer_related_offences.head()
```

	computer_related_offences_2008	computer_related_offences_2009	computer_related_offences_2010	computer_related_offences_2011	computer_relat
State/UT					
Andhra Pradesh	23	21	21	287	
Arunachal Pradesh	0	1	1	10	
Assam	1	2	2	25	
Bihar	0	0	0	19	
Chhattisgarh	0	2	2	0	

Regression models:

Two regression models are used for prediction, Gradient Boosting Regression and Ridge Regression.

Here, all cybercrime types from 2008 to 2019 are used for predicting that particular type of cybercrime for year 2020.

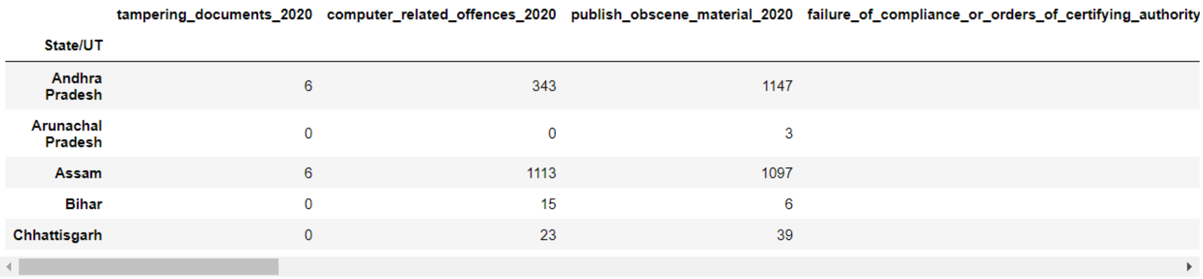
For applying model and getting prediction results the data is divided into standard 70:30 ratio, where 70 percent of the data is utilised for training purpose and 30 percent is used for testing the performance of the model. For this purpose, sklearn module model_selection's method train_test_split() is used. Train_test_split method gives divided data as X_train, X_test, y_train, y_test.

Next, to train the gradient boosting model x_train and y_train are fed to it, so that it learns the various relations among the variables. Then model predicts the values for y_test using it training understanding, x_test is used for prediction. For optimizing results, the parameters taken by the Gradient boosting model like learning rate, max depth, n_estimators are optimized as per data requirement.

The performance of the model is observed by root mean square error and mean absolute error using the predicted values (i.e., predicted with x_test values) and the actual values (i.e., y_test values). By minimising the values of root mean square error and mean absolute error the performance of the model is improved.

Predictions and Prediction analysis:

Using the gradient boosting model dataframe created for 2020 cybercrime values.



	tampering_documents_2020	computer_related_offences_2020	publish_obscene_material_2020	failure_of_compliance_or_orders_of_certifying_authority
State/UT				
Andhra Pradesh	6	343	1147	
Arunachal Pradesh	0	0	3	
Assam	6	1113	1097	
Bihar	0	15	6	
Chhattisgarh	0	23	39	

The predicted values for most vulnerable cybercrime in all states for 2020 is shown in the figure (4) below

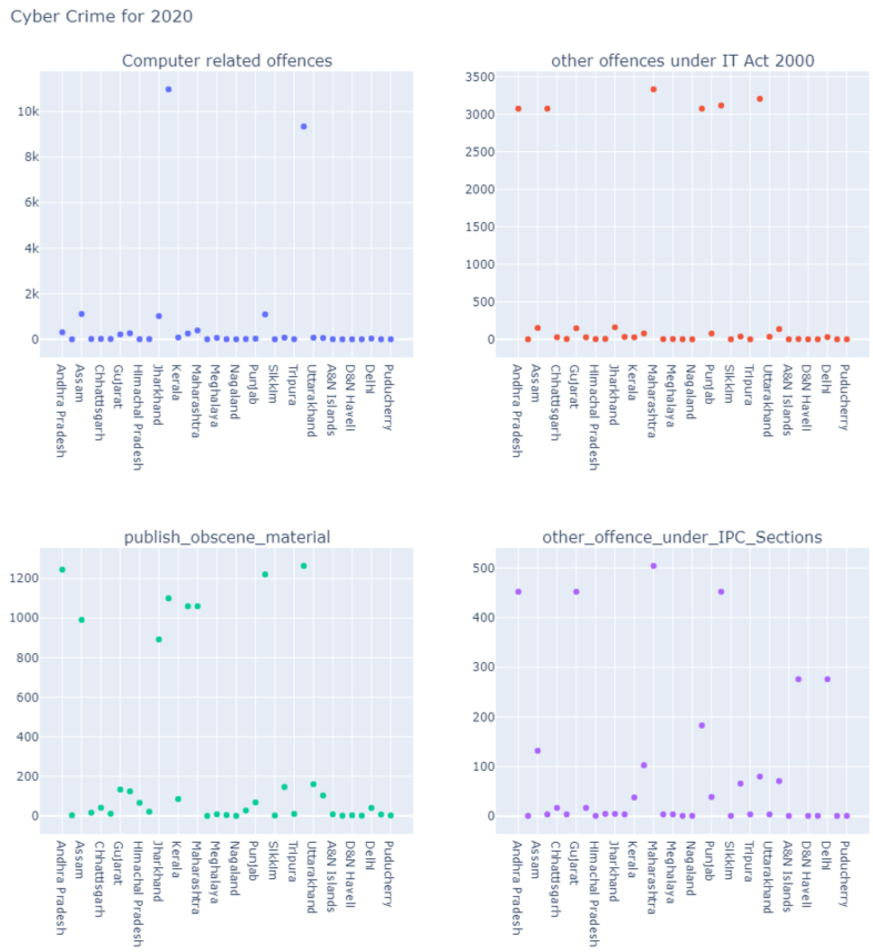
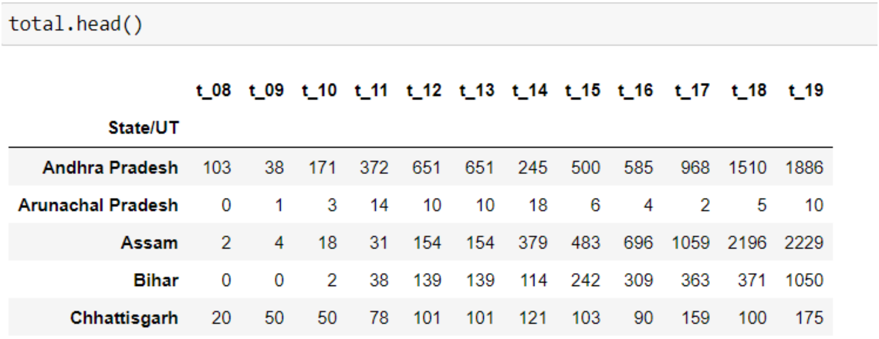


Figure (4)

Prediction analysis:

For cross checking the results of prediction:

Using the Gradient Boosting model, and taking the total cybercrime values of each year for States and Union territories the total cybercrime value for 2020 is predicted.

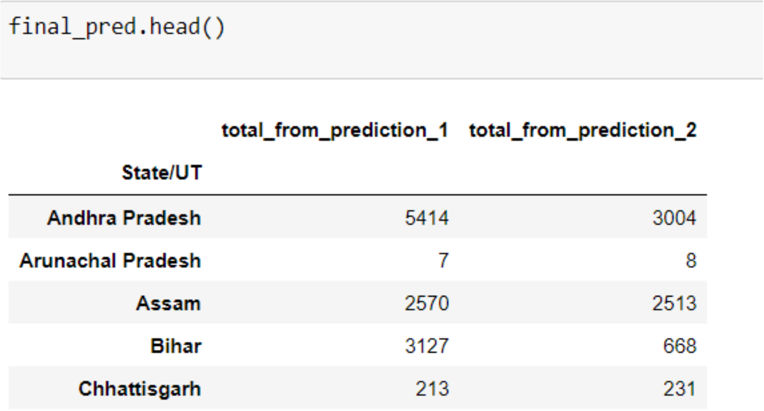


```
total.head()
```

State/UT	t_08	t_09	t_10	t_11	t_12	t_13	t_14	t_15	t_16	t_17	t_18	t_19
Andhra Pradesh	103	38	171	372	651	651	245	500	585	968	1510	1886
Arunachal Pradesh	0	1	3	14	10	10	18	6	4	2	5	10
Assam	2	4	18	31	154	154	379	483	696	1059	2196	2229
Bihar	0	0	2	38	139	139	114	242	309	363	371	1050
Chhattisgarh	20	50	50	78	101	101	121	103	90	159	100	175

Figure (5)

Figure (5) shows the dataframe with columns containing total cybercrime value for a particular year from 2008 to 2019 using this dataframe and applying gradient boosting regression model the total cybercrime values for States and Union Territories is predicted for 2020. Refer to figure (6) dataframe's column total_prediction_2 for predicted value obtained by this method.



```
final_pred.head()
```

State/UT	total_from_prediction_1	total_from_prediction_2
Andhra Pradesh	5414	3004
Arunachal Pradesh	7	8
Assam	2570	2513
Bihar	3127	668
Chhattisgarh	213	231

Figure (6)

Figure (6) shows the predicted values of cybercrime for 2020 using two methods, first by predicting the values for each cybercrime type and then using pandas for computing the total cybercrime for States and Union Territories, while second by predicting values for total cybercrime in 2020 using dataframe shown in figure (5).

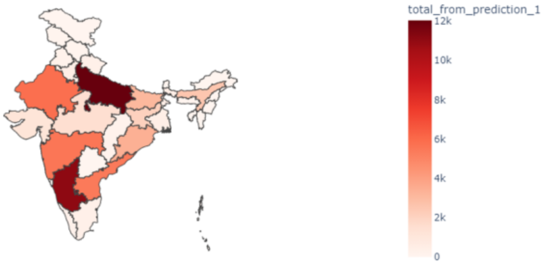


Figure (7)

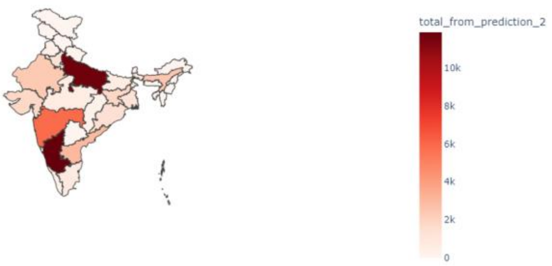


Figure (8)

figure (7) and figure (8) compare the values predicted for 2020 by two method discussed above.

Now, difference between two predictions as performed above can be observed by using sklearn module metric's methods mean_squared_error and mean_absolute_error.

```
mse = mean_squared_error(final_pred['total_from_prediction_1'], final_pred['total_from_prediction_2'])
rmse = np.sqrt(mse)
mae = mean_absolute_error(final_pred['total_from_prediction_1'], final_pred['total_from_prediction_2'])


print("root mean squared error : ",rmse)
print("mean absolute error : ", mae)
```

```
root mean squared error :  948.8371379144654
mean absolute error :  475.34285714285716
```