Combat Sexual Abuse via Machine Learning

Vyoma Patel[1], Bhavin Gor[2], Preyanshu Sukhadia[3] , Prasham Mody[3]

[1]vyoma.p@ahduni.edu.in
[2]bhavin.g@ahduni.edu.in
[3]preyanshu.s@ahduni.edu.in
[4]prahsam.m@ahduni.edu.in

Abstract— **One of the most sensitive and major concerns across globe-Women Safety. The early identification of the crucial issue of sexual harassment is vital for mitigating the risk. This can certainly be worked upon using Machine Learning techniques. The model aims to use classification algorithms based on abusing categories with the prediction of most prevailing locations. This data can be helpful for the government for taking necessary actions, safeguarding the nation and also alert females from such unavoidable incidents.**

Keywords— **sexual harassment | machine learning | classification | analysis | prevention | violence**

I. Introduction

The proposed project definition quests about how machine learning can provide new ways to overcome this gender-based violence.

This is one of the crucial problems which needs to be taken care of and hence necessary actions should be taken.

Artificial Intelligence has been prevalent in today's era for handling and mitigating such criminal incidents.

The AI model is a collaborative technique by Omdena and Safecity project.

This model leverages open-sourced data as well as Safecity's database to build heat-maps and safe routes using Nearby Search, Directions API and Grid Coverage techniques.

This Safecity platform acts as a crowd-sourced place where individuals can share their personal stories of sexual assaults and rapes.

II. Literature Survey

The prior research led to the use of several NLP techniques to easily aggregate and classify the labels. Deep learning algorithms like CNN and RNN were used for building and generating single and multi-label classifications. Also, some textual data of Twitter was used for sentimental analysis like generating word clouds associated with sexual assault cases. This indeed helped to aware citizens. But the scope was limited, and a rather complex approach.

III. Implementation

Firstly, the dataset was acquired from the Safecity. After acquiring the dataset, the dataset was preprocessed, Exploratory Data Analysis was performed and statistical and graphical representations were carried out. Then labels were assigned to the categories so that they can be to apply different algorithms to the data. Most of the important data was in text form, hence it was necessary to implement TF-IDFvectorizer and CountVectorizer so that the data can be used for feature extraction. Then, the most correlated unigrams and bigrams were found from the description of each category which eased the process of classification of different categories based on description. Initially, the following classification algorithms i.e. LogisticRegression, RandomForestClassifier, LinearSVC, and MultinomialNB were implemented. After implementation, the accuracies obtained were not desirable. For dimensionality reduction, Principle Component Analysis was used but it didn't make significant impact on their accuracies. In order to improve the accuracies, hyper-parameter tuning was used which was achieved with the help of GridSearchCV. After, hyper-parameter tuning, improvement in the accuracies were observed. The training and the test data were resampled with the help of RepeatedStratifiedKFold and it was used with GridSearchCV for different models. These results showed improved accuracies compared to the results obtained using GridSearchCV only. For comparison of these accuracies, line graphs were plotted at different stages representing the accuracies of different models which helps to understand the effects of different methods.

IV. Results

After EDA, as a result of preprocessing the data, the irregularities from the dataset were removed.

11201	5883	Comments and stalking at Goregaon railway station	2017-09-29 00:00:00	Shakti Niwas, Station Rd, Jawahar Nagar, Goreg...	NaN	Commenting, Stalking,

Fig. 1 Data entries

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1310 entries, 0 to 1309
Data columns (total 40 columns):
 #   Column                                   Non-Null Count  Dtype
---  ------                                   --------------  -----
 0   #                                        1310 non-null   int64
 1   INCIDENT TITLE                           1310 non-null   object
 2   INCIDENT DATE                            1310 non-null   object
 3   LOCATION                                 1310 non-null   object
 4   DESCRIPTION                              1305 non-null   object
 5   CATEGORY                                 1310 non-null   object
 6   LATITUDE                                 1310 non-null   float64
 7   LONGITUDE                                1310 non-null   float64
 8   More Info                                0 non-null      float64
 9   YEAR                                     1310 non-null   int64
 10  MONTH                                    1310 non-null   int64
 11  DAY                                      1310 non-null   int64
 12  HOUR                                     1309 non-null   float64
 13  DAYOFWEEK                                1309 non-null   object
 14  Touching /Groping                        1309 non-null   float64
 15  Catcalls/Whistles                        1309 non-null   float64
 16  Sexual Invites                           1309 non-null   float64
 17  Stalking                                 1309 non-null   float64
```

```
 18  Others                                        1309 non-null   float64
 19  Commenting                                    1309 non-null   float64
 20  Rape / Sexual Assault                         1309 non-null   float64
 21  North East India Report                       1309 non-null   float64
 22  Indecent Exposure/Masturbation in public      1309 non-null   float64
 23  Chain Snatching                               1309 non-null   float64
 24  Ogling/Facial Expressions/Staring             1309 non-null   float64
 25  Taking pictures                               1309 non-null   float64
 26  Poor / No Street Lighting                     1309 non-null   float64
 27  Online Harassment                             1309 non-null   float64
 28  Human Trafficking                             1309 non-null   float64
 29  Petty Robbery                                 1309 non-null   float64
 30  NUMBER_CAT                                    1309 non-null   float64
 31  ADDRESS                                       1309 non-null   object
 32  POSITION                                      1309 non-null   object
 33  COUNTRY                                       1309 non-null   object
 34  STATE                                         1309 non-null   object
 35  COUNTY                                        1309 non-null   object
 36  LABEL                                         1309 non-null   object
 37  CITY                                          1308 non-null   object
 38  DISTRICT                                      1226 non-null   object
 39  STREET                                        541 non-null    object
dtypes: float64(21), int64(4), object(15)
memory usage: 409.5+ KB
```

Fig. 2 Structure of the dataset

Fig. 1 & 2 represents the structure of the dataset and the data entries before preprocessing.

11200	5884	Commenting, Stalking and etc at Goregaon	2017-10-03 10:00:00	Shakti Niwas, Station Rd, Jawahar Nagar, Goreg...	NaN	Stalking
11201	5883	Comments and stalking at Goregaon railway station	2017-09-29 00:00:00	Shakti Niwas, Station Rd, Jawahar Nagar, Goreg...	NaN	Commenting

Fig. 3 Data entries

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1310 entries, 0 to 1309
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   #               1310 non-null   int64
 1   INCIDENT TITLE  1310 non-null   object
 2   INCIDENT DATE   1310 non-null   object
 3   LOCATION        1310 non-null   object
 4   DESCRIPTION     1305 non-null   object
 5   CATEGORY        1310 non-null   object
 6   LATITUDE        1310 non-null   float64
 7   LONGITUDE       1310 non-null   float64
 8   DAYOFWEEK       1309 non-null   object
 9   NUMBER_CAT      1309 non-null   float64
 10  COUNTRY         1309 non-null   object
 11  STATE           1309 non-null   object
 12  COUNTY          1309 non-null   object
 13  CITY            1308 non-null   object
 14  DISTRICT        1226 non-null   object
dtypes: float64(3), int64(1), object(11)
memory usage: 153.6+ KB
```

Fig. 4 Structure of the dataset

Fig. 3 & 4 represents the structure of the dataset and the data entries after preprocessing.

From EDA and visualizations some interesting patterns in the data were observed.

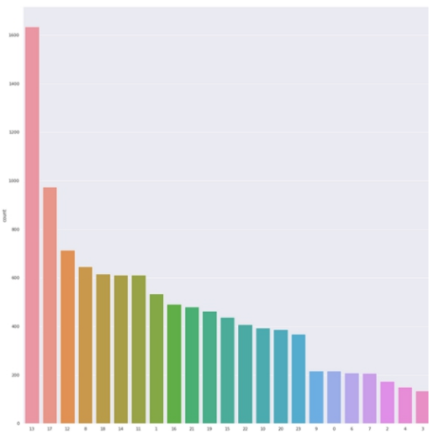

Fig. 5 Incident hour counts

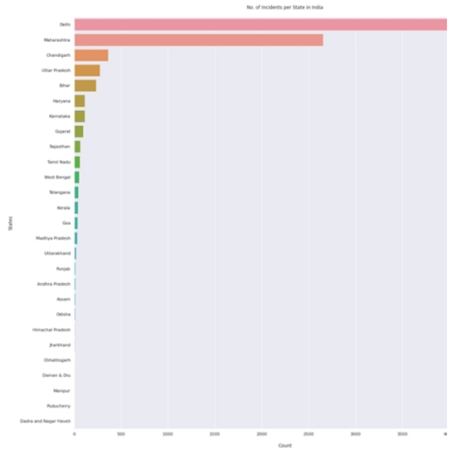

Fig. 6 Incident state counts

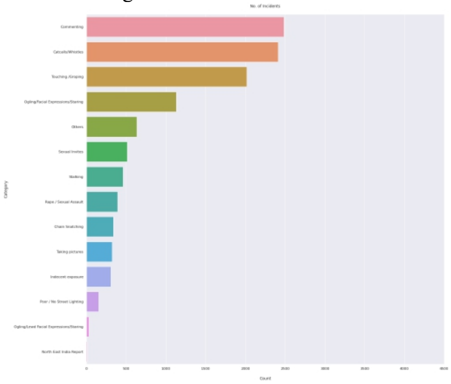

Fig. 7 Category counts

Fig. 5, 6 & 7 are some of the visualizations from the project. These are the bar graphs which represent the peak hours, the cities having majority cases and the most prevalent categories.

After applying various commonly used machine learning algorithms classification of the category from the description was possible.

```
'Catcalls/Whistles' classified as 'Touching /Groping' : 76 examples.
```

	CATEGORY	DESCRIPTION
6530	Catcalls/Whistles	touching , groping...
1384	Catcalls/Whistles	When I'm mostly travelling in buses, trains et...
8081	Catcalls/Whistles	while walking for the platform a guy crossed b...
5204	Catcalls/Whistles	Three years back when my friend and I were tra...
7714	Catcalls/Whistles	40+ more year old man tries to touch me while ...
...	...	...
5694	Catcalls/Whistles	My friend and I were walking back from class a...
7220	Catcalls/Whistles	Groping
7358	Catcalls/Whistles	i know a girl who has been harresed .she was b...
7605	Catcalls/Whistles	i was travelling in the public transport where...
1940	Catcalls/Whistles	i was feeling elpless

76 rows × 2 columns

```
'Sexual Invites' classified as 'Touching /Groping' : 16 examples.
```

	CATEGORY	DESCRIPTION
9653	Sexual Invites	in evening it was a rainy day i took auto but ...
11136	Sexual Invites	At the age of four or five, I went to my frien...
256	Sexual Invites	Some boys like embarasing girls if they refuse...
10415	Sexual Invites	This incident took place in the night.\nWhile ...

Fig. 8 Classification results

Fig. 8 represents the correctly classified cases of category from description.

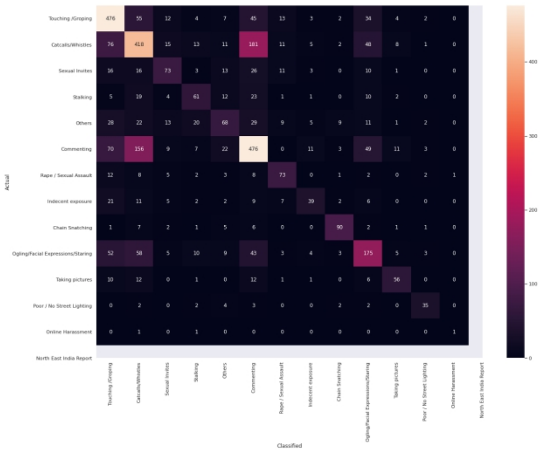

Fig. 9 Confusion matrix

Fig. 9 represents the confusion matrix depicting the actual vs the classified categories which helps in knowing the misclassified categories.

After implementing the classification algorithms, their accuracies were calculated in order to determine the best model. After hype-parameter tuning, the accuracies for some of the classification algorithms were found. Their accuracies and their differences are represented in the following figures (Fig. 10-16).

	model	best_score	best_params
0	svm	0.520279	{'C': 0.8, 'kernel': 'linear'}
1	random_forest	0.520261	{'n_estimators': 20}
2	logistic_regression	0.530169	{'C': 1, 'penalty': 'l2', 'solver': 'liblinear'}
3	naive_bayes_gaussian	0.292446	{}
4	naive_bayes_multinomial	0.476425	{}
5	decision_tree	0.434799	{'criterion': 'gini'}

Fig. 10 Models, their accuracies with their parameters

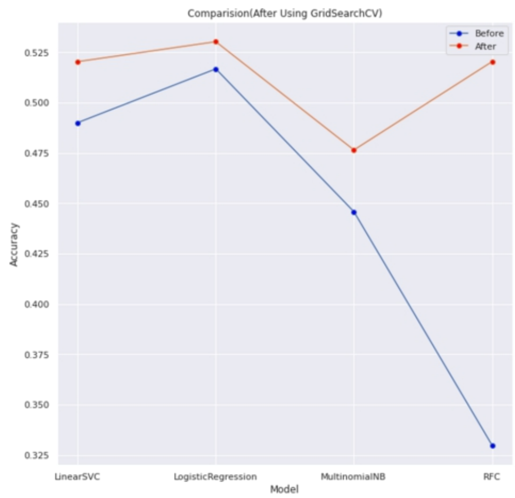

Fig. 11 Accuracy comparison for LinearSVC, Logistic Regression, MultinomialNB and RFC before and after using only GridSearchCV

	Accuracy	Parameter
0	0.488911	{'C': 100, 'penalty': 'l2', 'solver': 'newton-...
1	0.492933	{'C': 100, 'penalty': 'l2', 'solver': 'lbfgs'}
2	0.504236	{'C': 100, 'penalty': 'l2', 'solver': 'libline...
3	0.522149	{'C': 10, 'penalty': 'l2', 'solver': 'newton-cg'}
4	0.522516	{'C': 10, 'penalty': 'l2', 'solver': 'lbfgs'}
5	0.532346	{'C': 10, 'penalty': 'l2', 'solver': 'liblinear'}
6	0.533409	{'C': 1.0, 'penalty': 'l2', 'solver': 'newton-...
7	0.533409	{'C': 1.0, 'penalty': 'l2', 'solver': 'lbfgs'}
8	0.530128	{'C': 1.0, 'penalty': 'l2', 'solver': 'libline...
9	0.385185	{'C': 0.1, 'penalty': 'l2', 'solver': 'newton-...
10	0.385185	{'C': 0.1, 'penalty': 'l2', 'solver': 'lbfgs'}
11	0.387016	{'C': 0.1, 'penalty': 'l2', 'solver': 'libline...
12	0.216890	{'C': 0.01, 'penalty': 'l2', 'solver': 'newton...
13	0.216890	{'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
14	0.217981	{'C': 0.01, 'penalty': 'l2', 'solver': 'liblin...

Fig. 12 Logistic Regression accuracies with different parameters

	Accuracy	Parameter
0	0.545493	{'alpha': 1.0}
1	0.546226	{'alpha': 1.1}
2	0.546592	{'alpha': 1.2}
3	0.552062	{'alpha': 1.5}
4	0.553165	{'alpha': 1.6}
5	0.553528	{'alpha': 1.8}
6	0.553161	{'alpha': 1.9}

Fig. 13 Ridge Classifier accuracies with different parameters

	Accuracy	Parameter
0	0.342849	{'C': 50, 'gamma': 'scale', 'kernel': 'poly'}
1	0.520656	{'C': 50, 'gamma': 'scale', 'kernel': 'rbf'}
2	0.444724	{'C': 50, 'gamma': 'scale', 'kernel': 'sigmoid'}
3	0.466977	{'C': 50, 'gamma': 'scale', 'kernel': 'linear'}
4	0.416945	{'C': 10, 'gamma': 'scale', 'kernel': 'poly'}
5	0.520656	{'C': 10, 'gamma': 'scale', 'kernel': 'rbf'}
6	0.472846	{'C': 10, 'gamma': 'scale', 'kernel': 'sigmoid'}
7	0.481589	{'C': 10, 'gamma': 'scale', 'kernel': 'linear'}
8	0.433405	{'C': 1.0, 'gamma': 'scale', 'kernel': 'poly'}
9	0.517702	{'C': 1.0, 'gamma': 'scale', 'kernel': 'rbf'}
10	0.523925	{'C': 1.0, 'gamma': 'scale', 'kernel': 'sigmoid'}
11	0.529392	{'C': 1.0, 'gamma': 'scale', 'kernel': 'linear'}
12	0.213959	{'C': 0.1, 'gamma': 'scale', 'kernel': 'poly'}
13	0.215424	{'C': 0.1, 'gamma': 'scale', 'kernel': 'rbf'}
14	0.314011	{'C': 0.1, 'gamma': 'scale', 'kernel': 'sigmoid'}
15	0.319474	{'C': 0.1, 'gamma': 'scale', 'kernel': 'linear'}
16	0.216890	{'C': 0.01, 'gamma': 'scale', 'kernel': 'poly'}
17	0.216890	{'C': 0.01, 'gamma': 'scale', 'kernel': 'rbf'}
18	0.216890	{'C': 0.01, 'gamma': 'scale', 'kernel': 'sigmo...
19	0.216890	{'C': 0.01, 'gamma': 'scale', 'kernel': 'linear'}

Fig. 14 SVC accuracies with different parameters

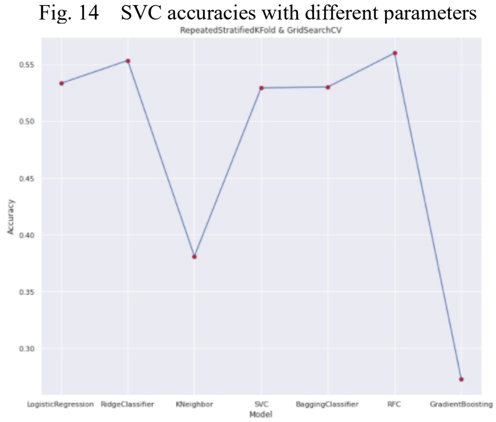

Fig. 15 Accuracies for different algorithms after using RepeatedStratifiedKFold and GridSearchCV

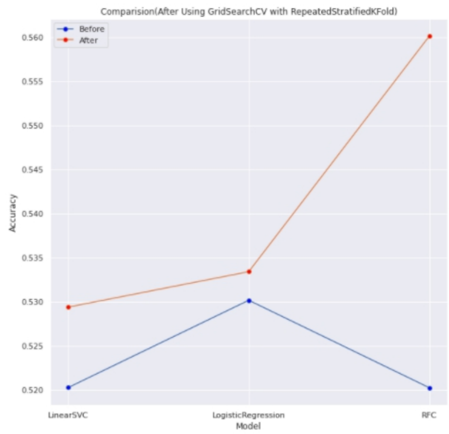

Fig. 16 Accuracy comparison for different algorithms after using RepeatedStratifiedKFold and GridSearchCV

V. Conclusion

Considerably less accuracy was obtained from all the applied machine learning algorithms. Almost half of the results lead to faulty classification which needs to be worked upon to get a better outcome. After implementing other features like PCA, hyper-parameter tuning and trying new models, the accuracy was improved but this improvement was not much significant.

Future work : Using common machine learning algorithms with GridSearchCV for hyper-parameter tuning is time consuming and the results obtained are not much accurate. To solve this problem, deep learning can be used which is fast and provides more accurate results.

Stemming and Lemmatization can be used for Natural Language Processing.

SMOTE (Synthetic Minority Oversampling Technique) can be used to resolve the imbalance problem in the dataset.

References

[1] Karlekar, S. and Bansal, M., 2018. *SafeCity: Understanding Diverse Forms of Sexual Harassment Personal Stories*. [online] Aclweb.org. Available at: <https://www.aclweb.org/anthology/D18-1303.pdf> [Accessed 17 March 2021].

[2] CHANDRA, V. and SRINATH, R., 2020. *Analysis of Women Safety using Machine Learning on Tweets*. [online] Irjet.net. Available at: <https://www.irjet.net/archives/V7/i6/IRJET-V7I6427.pdf> [Accessed 17 March 2021].

[3] Pandey, R., Purohit, H., Stabile, B. and Grant, A., 2018. *Distributional Semantics Approach to Detect Intent in Twitter Conversations on Sexual Assaults*. [online] Arxiv.org. Available at: <https://arxiv.org/pdf/1810.01012.pdf> [Accessed 17 March 2021].

[4] D, H., 2021. *ANALYSIS OF WOMEN SAFETY IN INDIAN CITIES USING DEEP LEARNING ON TWEETS*. [online] Ijsdr.org. Available at: <https://www.ijsdr.org/viewpaperforall.php?paper=IJSDR2006070> [Accessed 17 March 2021].

[5] Sánchez-Medina, A., Galván-Sánchez, I. and Fernández-Monroy, M., 2020. *Applying artificial intelligence to explore sexual cyberbullying behaviour*. [online] sciencedirect.com. Available at: <https://www.sciencedirect.com/science/article/pii/S2405844020300633> [Accessed 17 March 2021].

[6] Chu, T., Jue, K. and Wang, M., 2017. *Comment Abuse Classification with Deep Learning*. [online]Web.stanford.edu. Available at: <https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2762092.pdf> [Accessed 17 March 2021].

[7] Yadav, B., Sheshikala, M., Swathi, N., Chythanya, K. and E, S., 2020. *ShieldSquare Captcha*. [online] Iopscience.iop.org. Available at: <https://iopscience.iop.org/article/10.1088/1757-899X/981/2/022042/pdf> [Accessed 17 March 2021].

[8] Arora, I., 2021. *Document feature extraction and classification*. [online] Towards Data Science. Available at: <https://towardsdatascience.com/document-feature-extraction-and-classification-53f0e813d2d3> [Accessed 10 April 2021].

[9] PRANCKEVIČIUS, T. and MARCINKEVIČIUS, V., 2021. *Comparison of Naive Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression Classifiers for Text Reviews Classification*. [online] researchgate. Available at: <https://www.researchgate.net/publication/318056374_Comparison_of_Naive_Bayes_Random_Forest_Decision_Tree_Support_Vector_Machines_and_Logistic_Regression_Classifiers_for_Text_Reviews_Classification> [Accessed 10 April 2021].