

# Domestic Violence Detection from Social Media Content

## Table of Contents

1. INTRODUCTION.....	2
2. DATA ACQUISITION.....	2
3. DATA CLEANING AND.....	2
4. NLP PREPROCESSING.....	3
5. TOPIC MODELING.....	3
6. MERICS AND PREPROCESSING BEFORE MODELING.....	8
6.1 Define evaluation metrics	
6.2 Train test split	
6.3 Vectorization and scaling	
7. MODELING AND EVALUATION.....	9
7.1 Traditional machine learning algorithms	
7.1.1 Logistic Regression	
7.1.2 Random Forest	
7.1.3 SVM	
7.1.4 Naive Bayes	
7.1.5 XGBoost	
7.1.6 Best Performing Model	
7.2 Deep leaning algorithm	
8. RESULTS & NEXT STEPS.....	13
8.1 Results	
8.2 Next steps	
9. SUMMARIZATION ON DEALING WITH IMBALANCED DATA.....	14
9.1 Define imbalanced data	
9.2 Why It causes problems	
9.3 How to deal with imbalanced data	
9.3.1 Evaluation metrics	
9.3.2 Model-level methods	
9.3.3 Resampling methods	
9.3.4 Combine multiple methods	

## 1. INTRODUCTION

Do you know that 1 in 4 women and 1 in 7 men have been victims of severe physical violence by an intimate partner? And more than 1 in 3 women and 1 in 4 men have experienced either physical violence, rape, or stalking by an intimate partner in their lifetime?

Those are staggering numbers that many people aren't aware of. One of the biggest reasons is that victims often don't talk about it with friends and family, instead many of them go online to search for answers or just to tell their stories and pour out their feelings. That's why social media plays a major role to provide support and resources for victims. For instance, prompt national help hotline, nearby support groups, shelters, etc.

This project focus on detecting domestic violence posts.

## 2. DATA ACQUISITION

Data was pulled from Reddit using the Pushshift application program interface. Public posts were collected from the domestic violence subreddit.

As a control group, two other subreddits focusing on relationships, which are closely related to domestic violence, were selected; four other non-related topics (fitness, jokes, meditation, studying) were also included in the dataset to ensure the domestic violence detection model will generalize well.

Group	Subreddit	Number of posts	Description
Domestic violence	Domestic violence	10279	Subreddit to discuss domestic violence
Non-domestic violence (Control)	Fitness	1541	A place for the pursuit of physical fitness goals
	Jokes	116221	jokes subreddit
	Meditation	10096	Community for sharing meditation experiences
	Studying	11843	A subreddit is for all things studying
	Relationships	93747	A place to discuss interpersonal relationship advice

Table 1. Summary of the collected data (after cleaning) from Reddit.

## 3. DATA CLEANING and EDA

- In the 'selftext' feature, there were many '[removed]' value, which were essentially users removed their posts sometime after they posted it. So all rows where this feature were '[removed]' were dropped.
- 'all\_text' feature were generated by concatenating title (which is the title of the posts) and 'selftext' (which is the content of the posts) together.
- There were two time features, 'created\_utc' and 'created', which are Coordinated Universal Time and local unix time respectively. Since local time makes more sense, 'created\_utc' column was deleted. Then local time was converted to pandas datetime.
- the 'hour' feature was generated from the datetime, under the supposition that victims might have a preferred posting time; for example, maybe they post more often at night, when their abuser is asleep.
- Numeric features were explored by plotting KDE plot and heatmap, there is no obvious correlation as shown below.

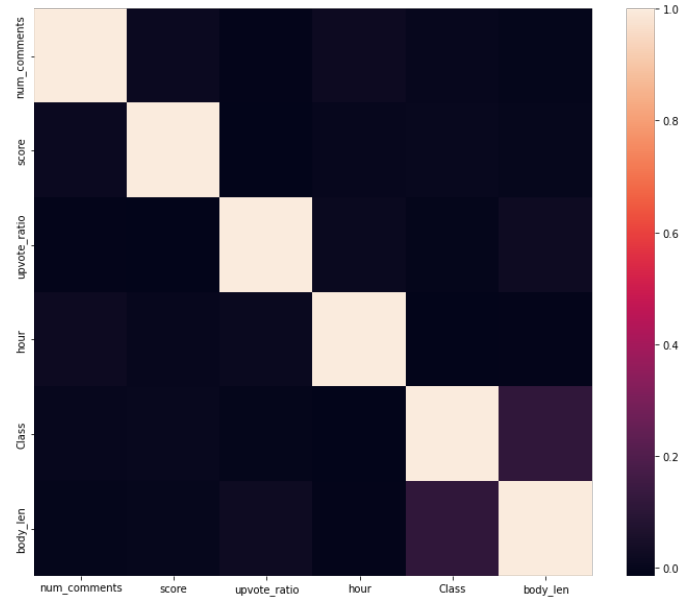


Figure 3.1. heatmap of the feature

## 4. NLP PREPROCESSING

For LDA and TF-IDF models, texts needed to be preprocessed.

- Convert emojis to text from `'I love Python ❤️, it is brilliant 🙌'` to `'I love Python :red_heart:, it is brilliant :thumbs_up:'`
- Using language-detection library langdetect to detect what languages are in the dataset, there were other languages other than English in the dataset, but they only comprised less than 1% of the data, they were dropped.
- Remove stopwords. Combined nltk and spacys stopwords together, and added custom stopwords, like 'said', 'say', 'tell', 'told', 'dont', 'got', 'im', 'know', 'knew', etc.
- Using a costume function to clean and tokenize text, converted words to lower case, removed punctuation and numbers, then used word\_tokenize from nltk to tokenize the text.
- Lemmatization. Lemmatization was chosen over stemming because lemmatization is a more sophisticated way to convert a word to its meaningful base. Stemming is a process that stems or removes last few characters from a word, often leading to incorrect meanings and spelling. A custom function was defined to lemmatize nouns, verbs, adjectives, and adverbs altogether. The default is to only lemmatize nouns.

## 5. TOPIC MODELING

Topic modeling is a type of statistical modeling for discovering the abstract “topics” that occur in a collection of documents. LDA(Latent Dirichlet Allocation) is one of several existing algorithms used to perform topic modeling in Python. Both Gensim and sklearn library were used to perform LDA. Gensim gave better results, it captured domestic violence topics every time while sklearn failed most of time. Both unigram and bigram were experimented in sklearn,

didn't work well. NMF(Non-Negative Matrix) in sklearn, on other hand gave a better result, and it was much faster. Non-Negative Matrix Factorization is a statistical method that helps us to reduce the dimension of the input corpora. Internally, it uses the factor analysis method to give comparatively less weightage to the words that are having less coherence.

Domestic Violence	
Topics	Words
family	mom, help, dad, family, year, work, mother, home, house, house
time	start, come, time, hit, try, night, say, ask, day, face
violence & abuse	violence, domestic, police, order, abuser, abuse, victim, woman, amp
crime	charge, arrest, jail, assault, neck, press, injury, court, Officer, prison
feel/Relationship	like, feel, want, Time, thing, relationship, Ive, year, think, love

Non Domestic Violence	
Topics	Words
study/school	study, learn, write, people, focus, subject, life, online, university, best
study	use, school, student, list, complete, test, study, task, grade, course
meditation	breath, breathing, pen, platform, store, khan, drug, clearly, participate, brother
feel	Feel: like, time, help, thing, want, day, exam, feel, need, start
fitness	thread, accountability, post, video, accountable, well-educated, versatile, toptictask, expertise, daily



The following graphs generated with pyLDAvis are the visualization of the topics:

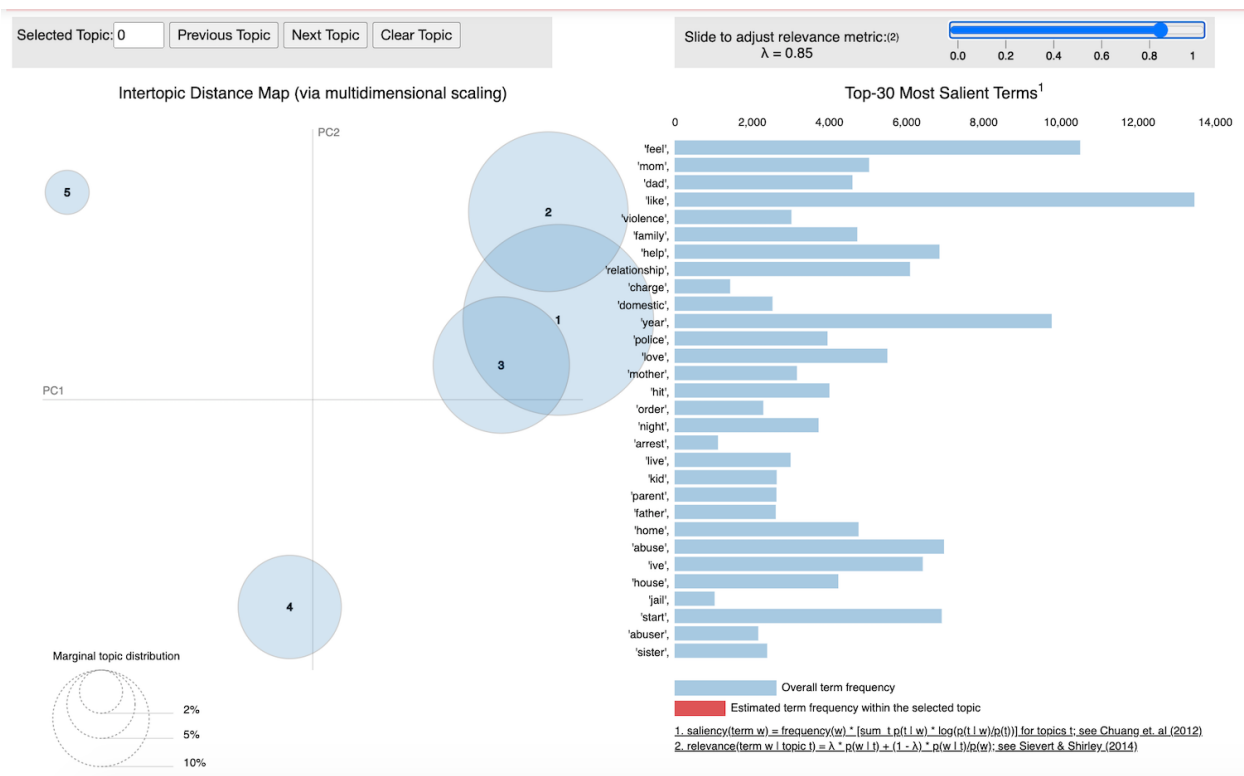


Figure 5.2 Top 30 Most Salient Terms of Domestic Violence Group

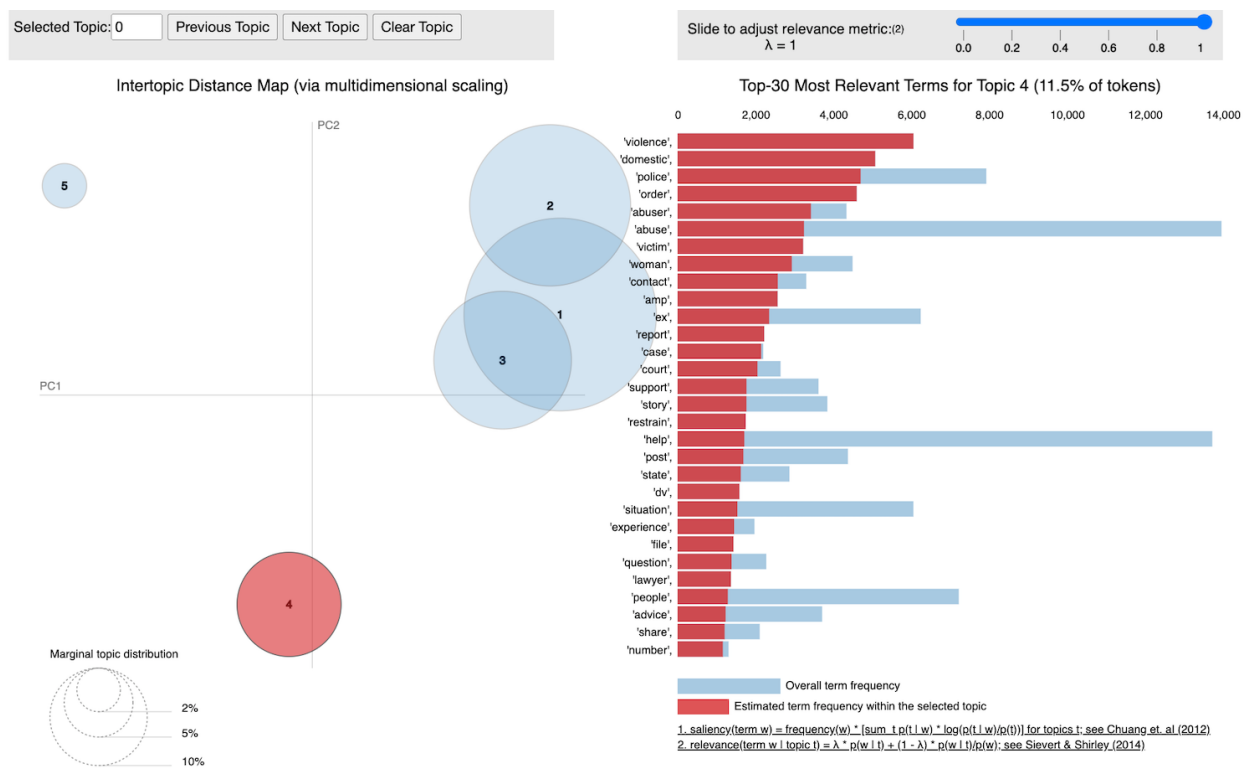


Figure 5.3 Top 30 Most Salient Terms for topic 4 of Domestic Violence Group

- Each bubble represents a topic. The larger the bubble, the higher percentage of the number of tweets in the corpus is about that topic.
- Blue bars represent the overall frequency of each word in the corpus. If no topic is selected, the blue bars of the most frequently used words will be displayed.
- Red bars give the estimated number of times a given term was generated by a given topic. As you can see from the image below, there are about 14,000 of the word 'abuse', and this term is used about 3,000 times within topic 4. The word with the longest red bar is the word that is used the most by the posts belonging to that topic.
- The further the bubbles are away from each other, the more different they are. For example, there are a lot of overlap words between topics 1 and 2. They seem to be both about feel and relationship, but it is much easier to tell the difference between topics 2 and 4. We can tell that topic 4 is about abuse.

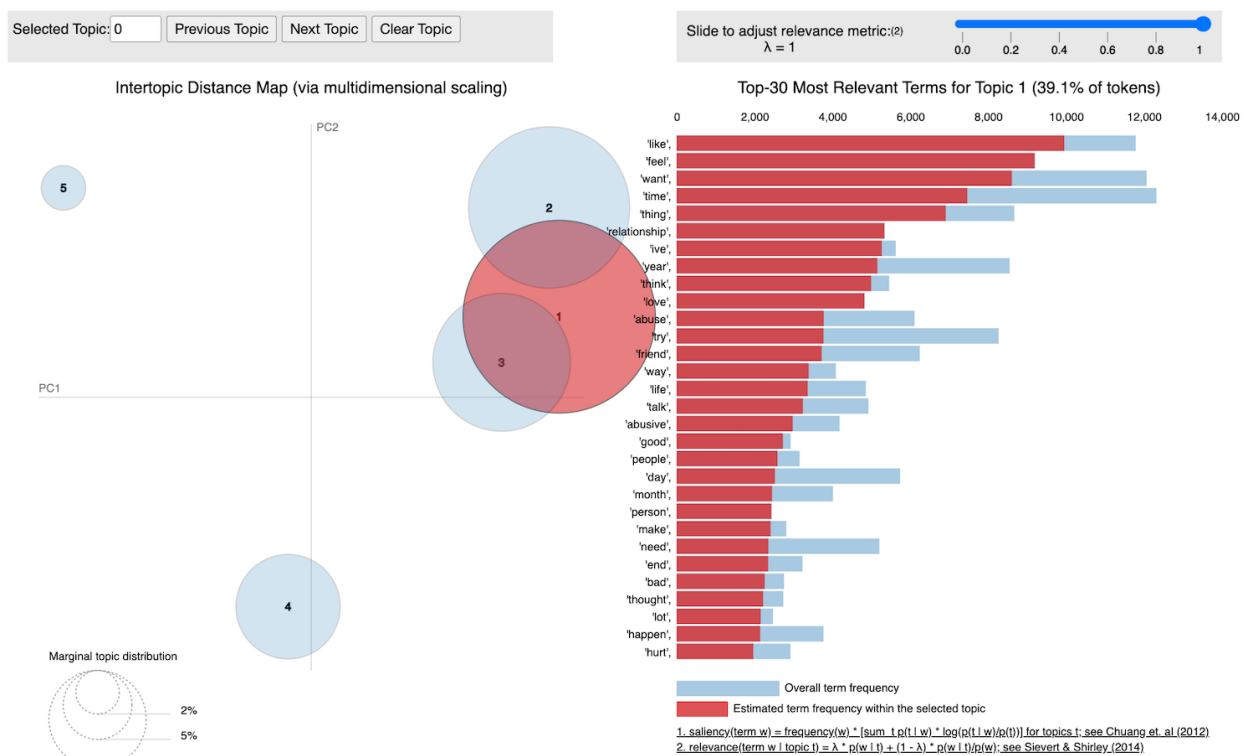


Figure 5.4 Top 30 Most Salient Terms for topic 1 of Domestic Violence Group

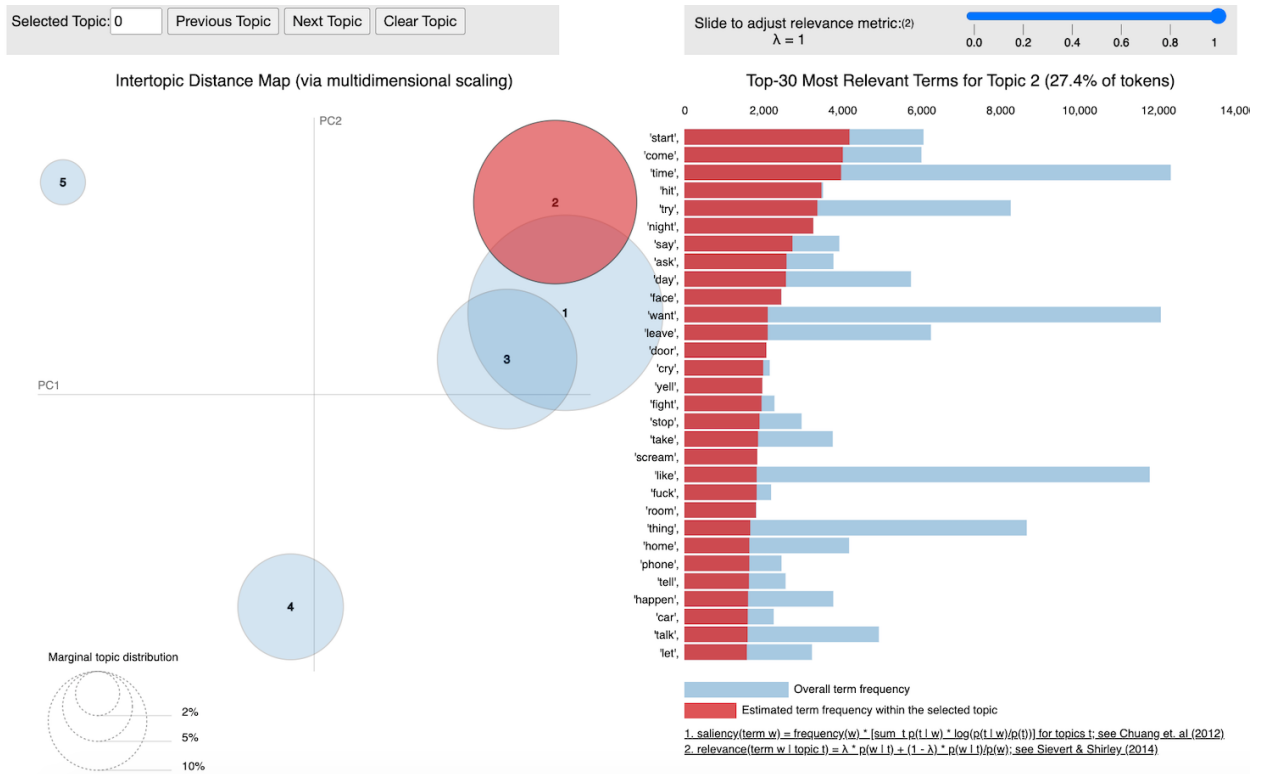


Figure 5.5 Top 30 Most Salient Terms for topic 2 of Domestic Violence Group

The following graph demonstrate the topics and words for non-domestic violence group:

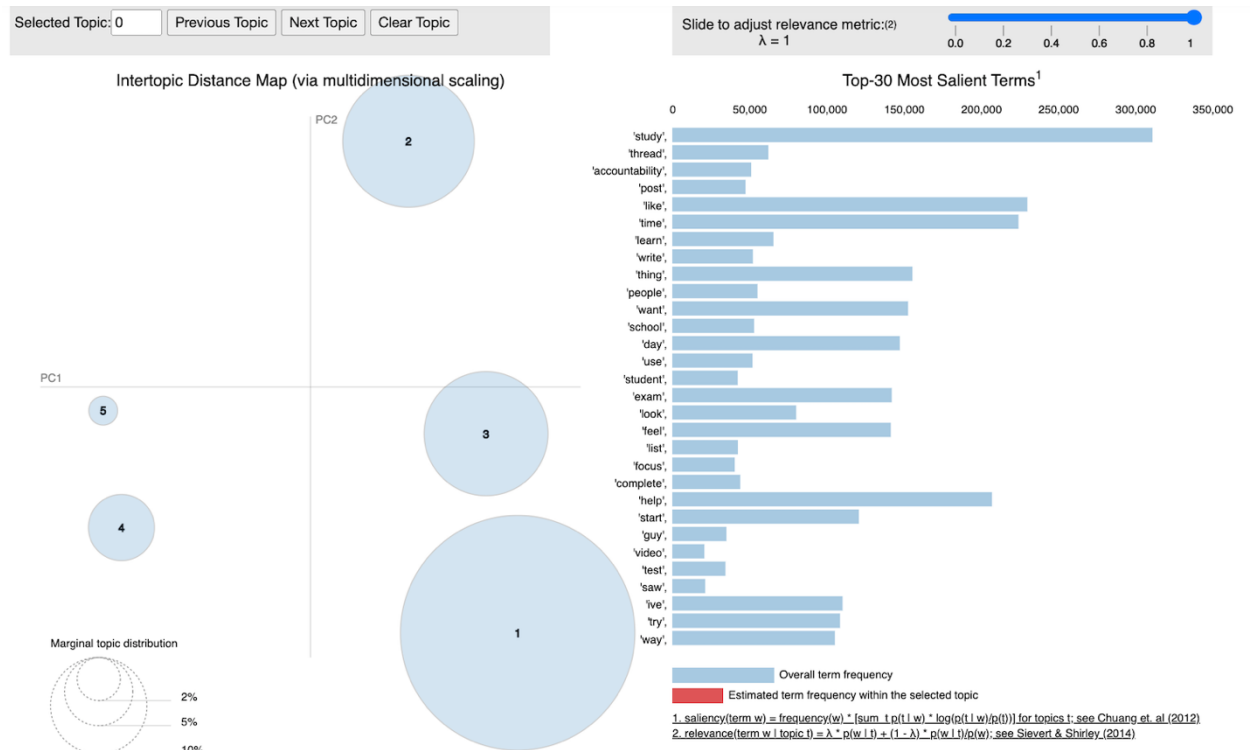


Figure 5.6 Top 30 Most Salient Terms of Non Domestic Violence Group

## 6. MERICS AND PREPROCESSING BEFORE MODELING

### 6.1 Define evaluation metrics

The data set is extremely imbalanced, positive class only comprise less than 5% of the whole dataset. And the goal is to detect domestic violence posts so that some measures can be taken to help the victims, the model should have a high recall and reasonable precision on the positive class. Hence classification metric like accuracy or ROC-AUC are not the right metrics. PRC (precision-recall curve) and F1 score are more appropriate metrics.

Moreover, classification report was generated for each model to inspect performance of each class.

### 6.2 Train test split

It is essential to stratify when train test split to preserves the same proportions of examples in each class as observed in the original dataset.

One approach to addressing the problem of class imbalance is to randomly resample the training dataset. The two main approaches to randomly resampling an imbalanced dataset are to delete examples from the majority class, called undersampling, and to duplicate examples from the minority class, called oversampling.


When using oversampling and undersampling method, it is important to make sure the validation set and test set are form the same distribution. Thus only the training set was oversampled or undersampled.

### 6.3 Vectorization and scaling

Text Vectorization is the process of converting text into numerical representation. In traditional machine learning algorithms, TfidfVectorizer and CountVectorizer were used.

CountVectorizer creates a matrix in which each unique word is represented by a column of the matrix, and each text sample from the document is a row in the matrix. The value of each cell is nothing but the count of the word in that particular text sample. This can be visualized as follows:

Data = ['The', 'quick', 'brown', 'fox', 'jumps', 'over', 'the', 'lazy', 'dog']



	The	quick	brown	fox	jumps	over	lazy	dog
Data	2	1	1	1	1	1	1	1

Figure 6.1 CountVectorizer

TF-IDF stands for Term Frequency Inverse Document Frequency of records. It can be defined as the calculation of how relevant a word in a series or corpus is to a text. The meaning increases proportionally to the number of times in the text a word appears but is compensated by the word frequency in the corpus (dataset). The mathematic concept is:



$$w_{x,y} = \text{tf}_{x,y} \times \log \left( \frac{N}{\text{df}_x} \right)$$

**TF-IDF**

Term  $x$  within document  $y$

$\text{tf}_{x,y}$  = frequency of  $x$  in  $y$   
 $\text{df}_x$  = number of documents containing  $x$   
 $N$  = total number of documents

Figure 6.2 TfidfVectorizer

In neural network, embedding layer was added. Using pre-trained word embedding GloVe and word2Vec word embedding trained on the domestic violence dataset were compared. They gave similar results.

Word2Vec is a technique used for learning word association in a natural language processing task. The algorithms in word2vec use a neural network model so that once a trained model can identify synonyms and antonyms words or can suggest a word to complete a partial incomplete sentence. Word2vec uses a list of numbers that can be called vectors to represent any distinct word. The cosine similarity between the vectors is used as the mathematical function for choosing the right vector which indicates the level of semantic similarity between the words. GloVe Embeddings are a type of word embedding that encode the co-occurrence probability ratio between two words as vector differences. GloVe is a combination of two words- Global and Vectors.

What is the difference between the two models? Word2vec embeddings are based on training a shallow feedforward neural network while glove embeddings are learnt based on matrix factorization techniques.

In practice, these two model give very similar results, which is also the case with this use case.

## 7. MODELING AND EVALUATION

### 7.1 Traditional machine learning algorithms

#### 7.1.1 Logistic Regression

The base model logistic regression with default hyperparameter setting gave a solid result; When changed the parameter class\_weight to 'balanced', give similar prc curve:

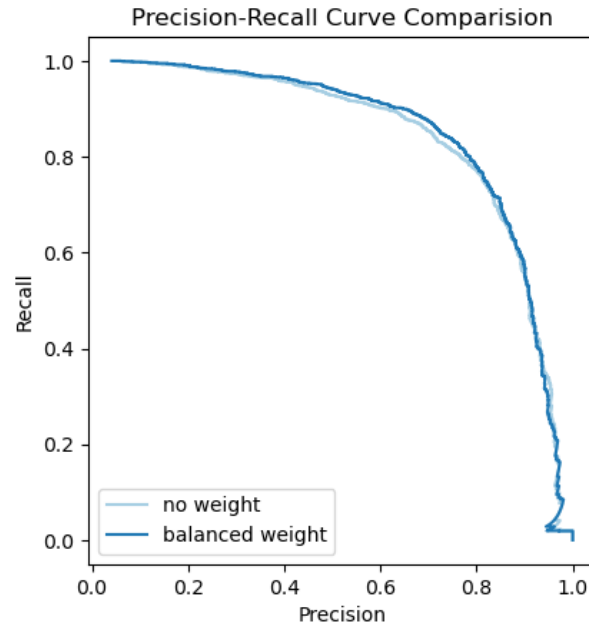


Figure 7.1 PRC comparison of logistic regression with vs without balanced weights

Then experimented with undersampling and oversampling methods, oversampling with example weights improves the score by a small amount and undersampling didn't work well.

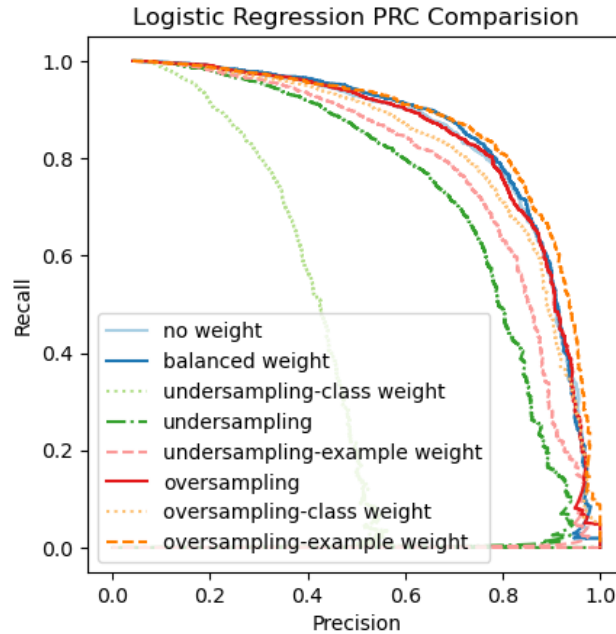


Figure 7.2 PRC comparison of logistic regression

### 7.1.2 Random Forest

Random forest didn't perform very well with the dataset. Although there are a couple of hyperparameters designed in the keras library to deal with imbalanced dataset, such as balanced class weights and balanced subsample class weights, they didn't help much.

RandomizedSearchCV was used to find the best hyperparameters, even the best one has pretty bad results, 0.40 recall and 0.25 precision.

Oversampling didn't work well too with 0.01 on positive class recall and 0.47 on positive class precision.

### 7.1.3 SVM

SVM gave good results:

	precision	recall	f1-score	support
0	0.99	0.99	0.99	56023
1	0.75	0.88	0.81	2466

However, the training time is too long, it took more than 8 hours to run, which is hundreds of times slower than logistic regression.

### 7.1.4 Naive Bayes

Naive Bayes models worked bad on the original dataset, but worked really well on oversampled datasets. The following is the prc comparison of the best performing model we discussed so far:

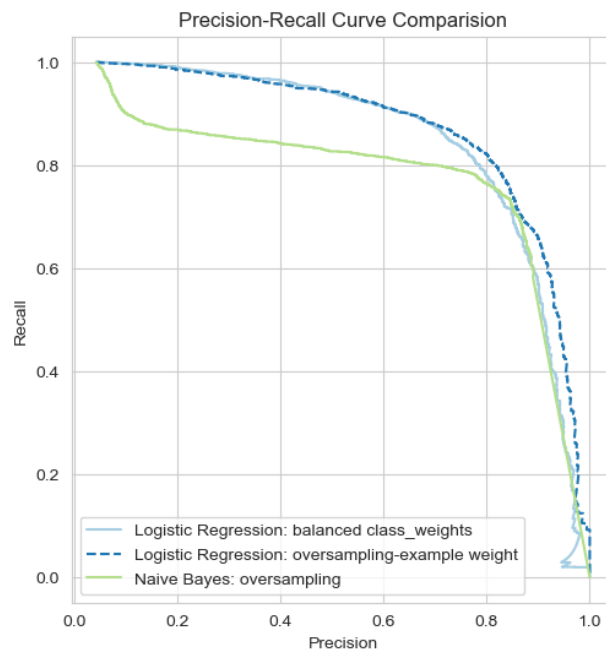


Figure7.3 PRC comparison of logistic regression and Naïve Bayes

### 7.1.5 XGBoost

XGBoost is a very powerful model, its performance with the default setting beat all the previous models with 0.94 precision, 0.89 recall and 0.91 f1-score on the positive class.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	42055
1	0.94	0.89	0.91	1812

And oversampling didn't improve performance.

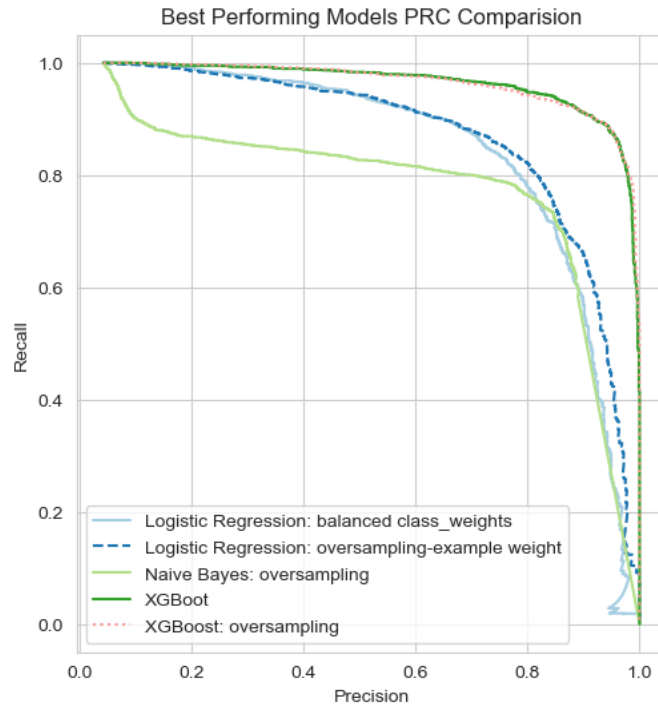


Figure7.4 PRC comparison of best performing models

## 7.2 Deep leaning algorithm

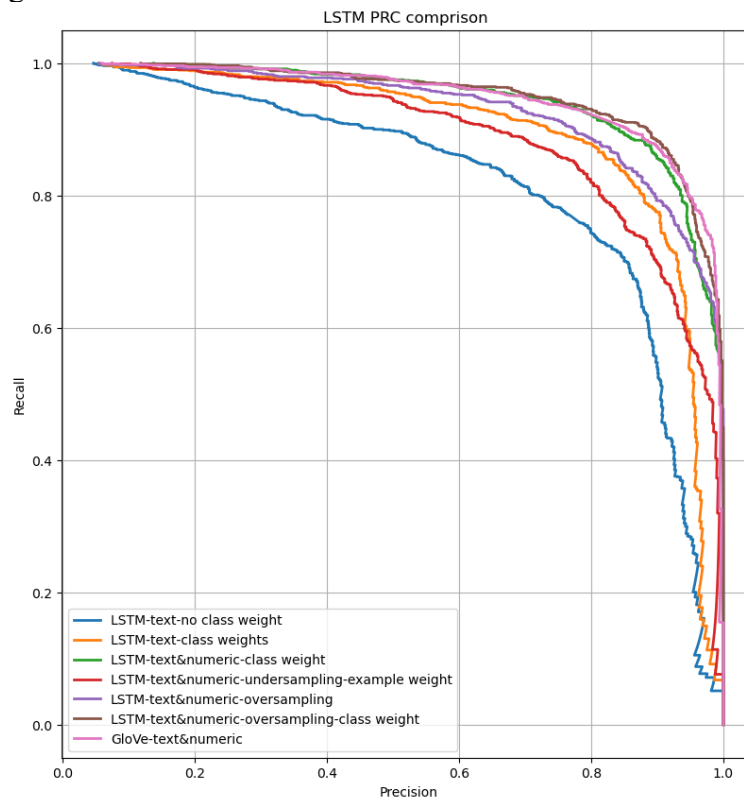


Figure7.5 PRC comparison of LSTM models

As the graph shows:

- Combine numeric features with text achieve better results than just text
- Class weight helped the performance for the original data but not so much with oversampled dataset
- Undersampling didn't work well while oversampling improved performance
- Undersampling with example weights worked better than just undersampling
- Generic LSTM and LSTM with GloVe gave similar results.

## 8. RESULTS & NEXT STEPS

### 8.1 Results

The best model performance are very similar with almost perfect scores for the negative class, and for the positive class, the scores vary depend on the threshold, and there is a tradeoff between precision and recall. If recall is around 0,95, then precision is around 0.6-0.7. Vice versa. If one want more balanced precision and recall score with a different threshold, both scores can be around 0.9. the following is the classification reports of XGBoost models at different thresholds:

Best F1-score:

	precision	recall	f1-score	support
0	0.99	1.00	1.00	42055
1	0.93	0.87	0.90	1812
accuracy			0.99	43867
macro avg	0.96	0.93	0.95	43867
weighted avg	0.99	0.99	0.99	43867

Best precision:

	precision	recall	f1-score	support
0	0.98	1.00	0.99	42055
1	0.96	0.63	0.77	1812
accuracy			0.98	43867
macro avg	0.97	0.82	0.88	43867
weighted avg	0.98	0.98	0.98	43867

Best recall:

	precision	recall	f1-score	support
0	1.00	0.99	0.99	42055
1	0.77	0.95	0.85	1812
accuracy			0.99	43867
macro avg	0.88	0.97	0.92	43867
weighted avg	0.99	0.99	0.99	43867

### 8.2 Next steps

- Try other word augmenter form nlpaug, eg. random word augmenter
- Try other augmenter form nlpaug, eg. sentence augmenter and character augmenter
- Try other augmentation methods, eg. SMOTE (synthetic minority oversampling technique) - creates synthetic examples of the rare class by combining original examples. It does this using a nearest neighbors approach.

- Try to improve the scores for positive class further
- Try to decrease the scores for negative class less than 1

## 9. SUMMARIZATION ON DEALING WITH IMBALANCED DATA

### 9.1 Define imbalanced data

An imbalanced dataset is a dataset where one or more labels make up the majority of the dataset, leaving far fewer examples of other labels. Classes that make up a large proportion of the data set are called majority classes. Those that make up a smaller proportion are minority classes. In this case: domestic violence posts is the minority class.

In many scenarios, getting more data for the minority class may be impractical or hard to acquire because the data is inherently imbalanced. e.g., fraud detection and detection of rare diseases.

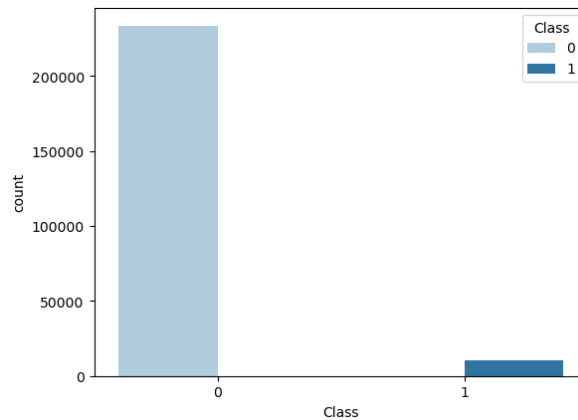


Figure 9.1 Class imbalance

### 9.2 Why it causes problems

With so few positives relative to negatives, the training model will spend most of its time on negative examples and not learn enough from positive ones.

An accuracy of over 90% can be misleading because the model may not have predictive power on the rare class if the majority class comprise over 90% of the dataset. Often, the minority class is more important than the majority class. A wrong prediction on an example of the minority class is more costly than a wrong prediction on an example of the majority class. For instance, missing a fraudulent transaction is 100x more costly than misclassifying a legitimate example as fraud.

### 9.3 How to deal with imbalanced data

#### 9.3.1 Evaluation metrics

- Accuracy is misleading when classes are imbalanced - performance of the model on the majority class will dominate the metric.  
Consider using accuracy for each class individually.
- Precision, recall, and F1 measure a model's performance with respect to the positive class in a binary classification problem.
- Precision-Recall curve (PRC) - identify a threshold that works best for the dataset. It gives more importance to the positive class (put emphasis on how many predictions the model got right out of the total number it predicted to be positive), which is helpful for

dealing with imbalanced data. And the AUC of PRC indicates model's predicting power.

### 9.3.2 Model-level methods

- Update loss function

Design a loss function that penalizes the wrong classifications of the minority class more than the wrong classifications of the majority class. Force the model to treat specific classes with more weight than others during training.

Most of the sklearn classifier modeling libraries and even some boosting based libraries like LightGBM and catboost have an in-built parameter "class\_weight" which helps optimize the scoring for the minority class .

By default, the value of class\_weight=None, i.e. both the classes have been given equal weights. Other than that, we can either give it as 'balanced' or we can pass a dictionary that contains manual weights for both the classes.

When the class\_weights = 'balanced', the model automatically assigns the class weights inversely proportional to their respective frequencies.

- Select appropriate algorithms

Logistic regression is able to handle class imbalanced relatively well in a standalone manner.

XGBoost works well with imbalanced dataset

Combine multiple techniques

### 9.3.3 Resampling methods

- Undersampling is the process where you randomly delete some of the observations from the majority class in order to match the numbers with the minority class.
- Oversampling is a little more complicated than undersampling. Oversampling is the process of adding more examples to the minority class. There are several ways, this can be achieved:
  - Random oversampling: Randomly make copies of the minority class until a ratio is reached.
  - Generate synthetic examples: It is the process of generating synthetic data that tries to randomly generate a sample of the attributes from observations in the minority class. There are a number of methods used to oversample a dataset for a typical classification problem. The most common technique is called SMOTE (Synthetic Minority Over-sampling Technique), which use a nearest neighbors approach.

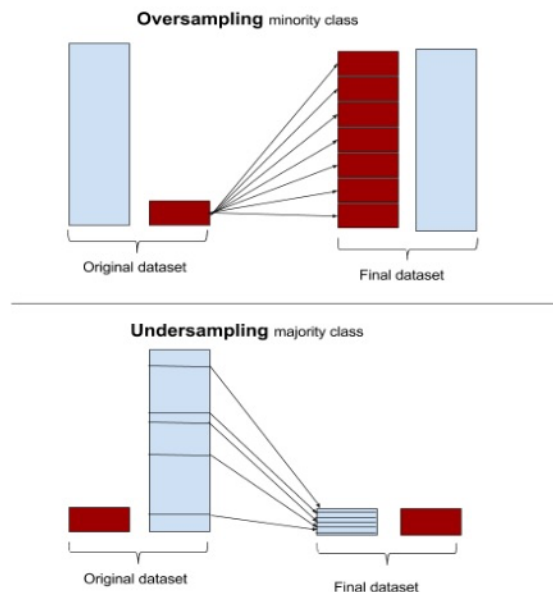


Figure 9.2 oversampling vs undersampling

- Example weight: The weight should be equal to the factor you used to downsample or upsampled, for example, in undersampling method:

$$\{\text{example weight}\} = \{\text{original example weight}\} \times \{\text{downsampling factor}\}$$

### 9.3.3 Data augmentation

There are many ways to do data augmentation, for instance there is an open source library called Imbalanced-learn. One can use SMOTE to generate synthetic data. For this project, nlpaug library was used for text augmentation. ContextualWordEmbsAug augmentaer was used, which applies word level operation to textual input based on contextual word embeddings. One can choose model type, commonly used models include BERT, RoBERTa, etc.

- Original text:

"ladies please take warning and stay aware from him  
<https://www.tiktok.com/t/ZTRPL4whN/> i'm trying to spread awareness to all so u don't go  
 through the physical and emotional abuse that he put me through.... ais there any advice to help  
 recover?"

- Augmented text:

"ladies please take warning and stay aware only from him <https://www.tiktok.com/t/ZTRPL4whN/> [UNK] i'm trying to spread awareness to all so [hope](#) u don't just go through the  
 physical and emotional abuse [spells](#) that he put his me through.... ais there any advice [here](#) to help  
 recover?"

Figure 9.3 augmentation example

The augmentation implemented did improve performance using undersampling method and no sampling, but didn't do much for oversampling method.



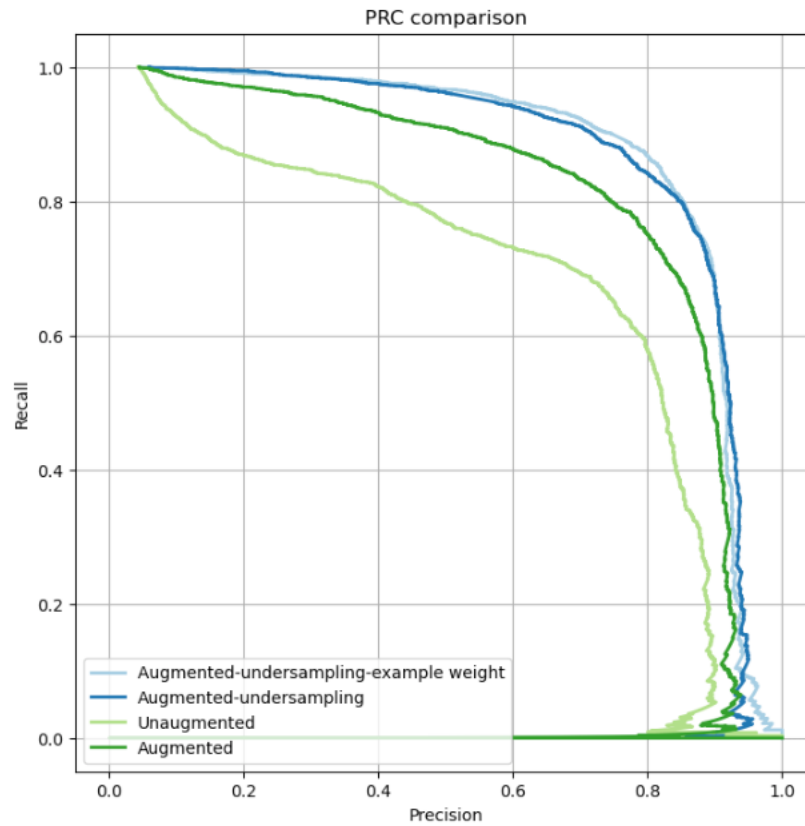


Figure 9.4 PRC comparison of LSTM Augmented vs Unaugmented

### 9.3.4 Combine multiple techniques

For example, Under-sampling + ensemble:

Use all samples of the minority class and a subset of the majority class to train multiple models and then ensemble those models.