Data Glacier
Your Deep Learning Partner

Hate Speech Detection

Final Report

Faris Chaudhry

09/06/23

Contents

Problem Statement

Assumptions

Data Cleaning and Pre-processing

EDA and Featurization

Models

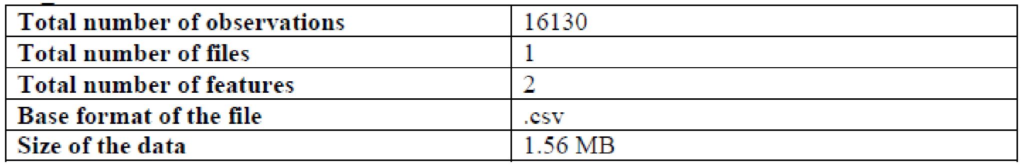
Business Understanding

- Hate Speech Detection is a task of sentiment classification.
- Censor hate speech posts.
    - These aren’t in line with our policy.
    - Defined as discriminatory messages based on identity.
- Earn user’s trust as safe and accessible platform.
- Raise advertiser confidence in brand image and platform.
    - Increase ad revenue.

Dataset and Assumptions

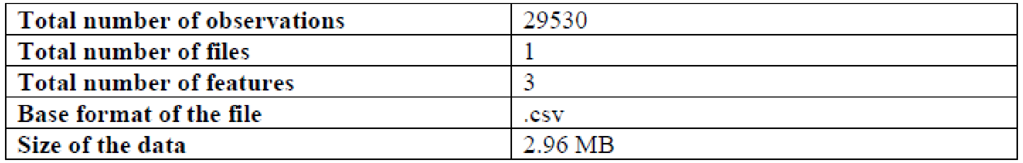
- The data is derived from real tweets.
- The training data is labelled correctly.
- The training and test data are from the same domain.
- The amount of hate speech compared to non-hate speech reflects the proportion on the platform (see below)..
- Tweets are below the limit of 200 characters and are formatted in the same way.

test_tweets.csv



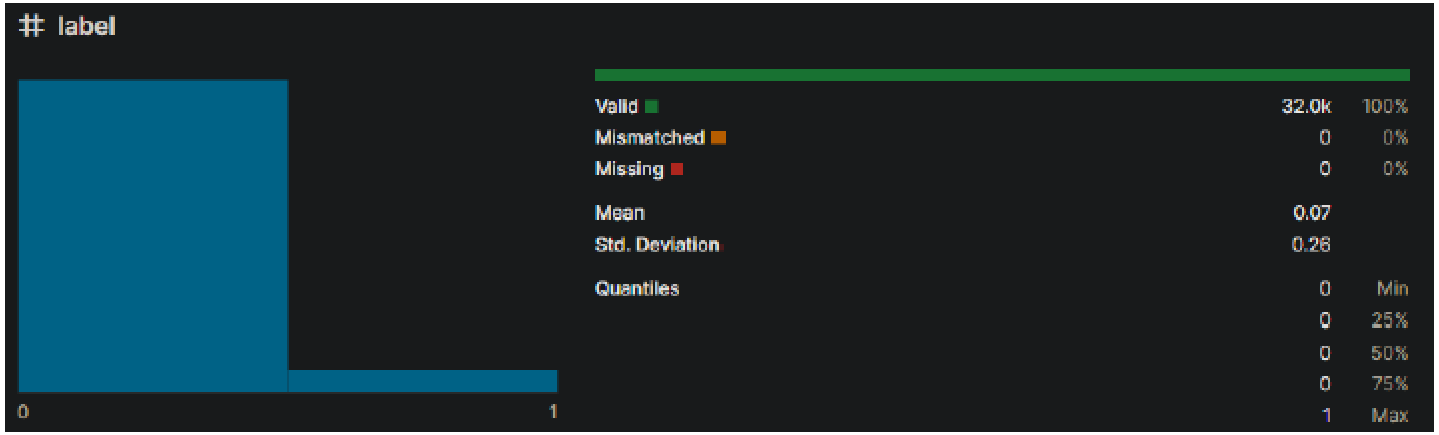
Total number of observations	16130
Total number of files	1
Total number of features	2
Base format of the file	.csv
Size of the data	1.56 MB

train.tweets.csv



Total number of observations	29530
Total number of files	1
Total number of features	3
Base format of the file	.csv
Size of the data	2.96 MB

Dataset split into training and testing data.

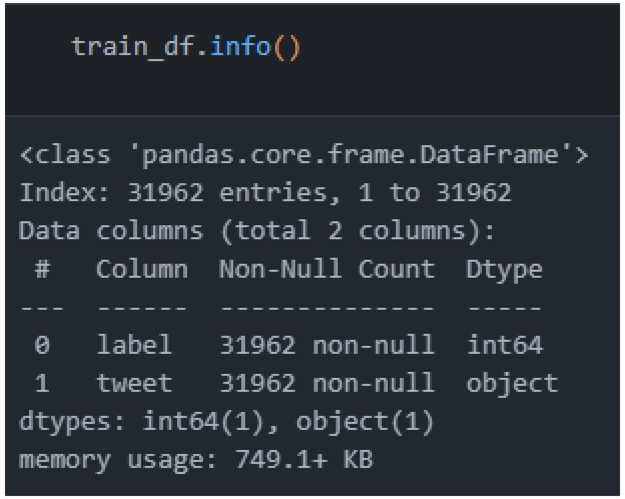


```
# label

Valid 🟩                 32.0k   100%
Mismatched 🟧                0     0%
Missing 🟥                   0     0%

Mean                      0.07
Std. Deviation            0.26

Quantiles                    0    Min
                             0    25%
                             0    50%
                             0    75%
                             1    Max
```

Data Pre-processing

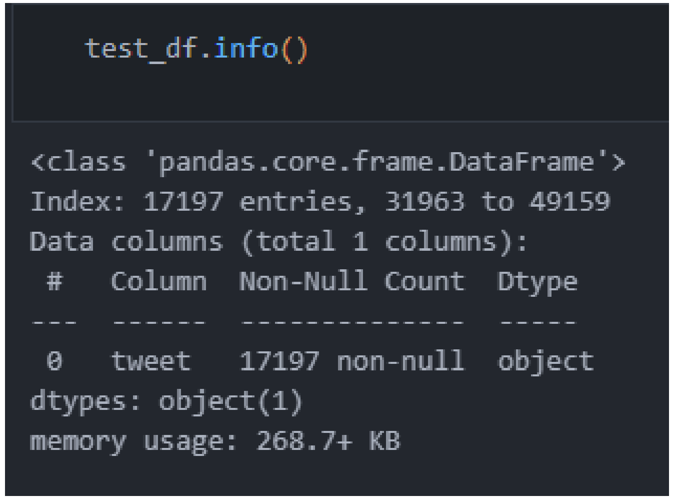
Column Validation

- Standardise Column Names
  - Lowercase
  - Replace spaces with underscores
- Remove duplicate tweets
- Remove null tweets
- Remove unlabelled training data
- Remove null and duplicate indexes



```
train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 31962 entries, 1 to 31962
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   label   31962 non-null  int64 
 1   tweet   31962 non-null  object
dtypes: int64(1), object(1)
memory usage: 749.1+ KB
```

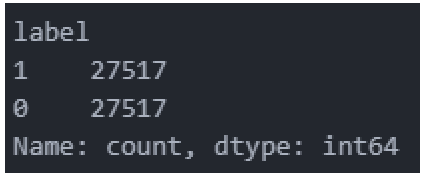


```
test_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 17197 entries, 31963 to 49159
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   tweet   17197 non-null  object
dtypes: object(1)
memory usage: 268.7+ KB
```

Up sampling

- Make number of samples labelled ‘0’ and ‘1’ same.
- Reduces bias towards randomly guessing ‘0’.
- Lowers false negatives.



```
label
1    27517
0    27517
Name: count, dtype: int64
```

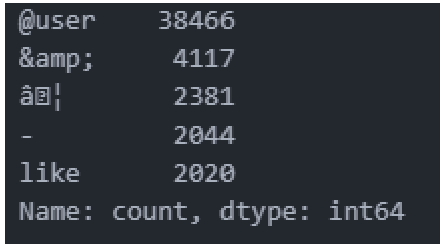
Data Cleaning

Data Cleaning (remove extra noise)

- Make all tweet words lowercase
- Remove punctuation
- Remove stop words (common words that add no information e.g., ‘the’, ‘and’, ‘a’, ‘I’)

Remove common and rare words and symbols.

- Common words show up too often.
    - Add extra dimensionality.
    - Don’t contain any information used for classification
- Rare words don’t have great enough sample size.
    - Overfitting to model
    - Add extra dimensionality
    - Around 200 tokens which only show up once in dataset.



```
@user     38466
&amp;      4117
â□¦         2381
-          2044
like       2020
Name: count, dtype: int64
```

Optional Data Cleaning

Spelling Correction

- Reduces lexicon of words which must be identified.
- Words may be corrected to the wrong word.
- Slang and language changed too quickly.

Suffix and Prefix Removal (or)
Lemmatization

- Reduces words to their root form.
- Reduces lexicon of words which must be identified.

Tokenization

- Tweets can be analysed by which their contained words.
- Loss: repetition of words and phrases is not considered.

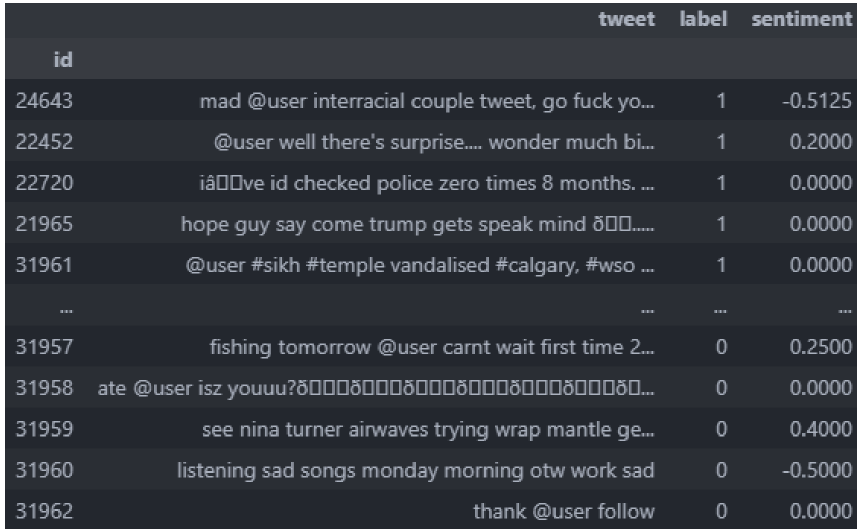
Vectorisation

- Each words gets its own vector.
- A tweet is identified by adding together the vectors of its words.
- Loss: adds extra noise and unnecessary dimensionality.
- Loss: doesn’t work if word is not found in model.

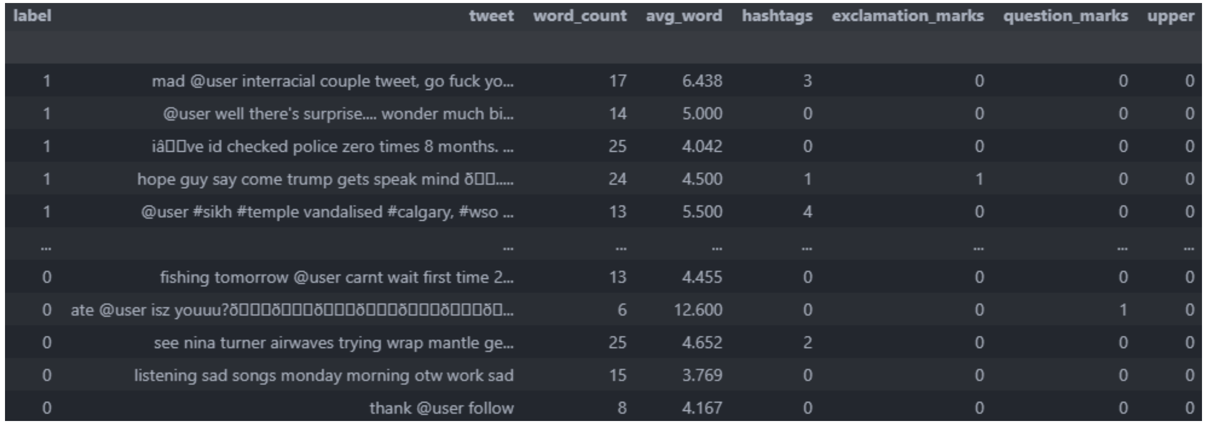
Extra Training Features

What factors are indicators of hate speech?

- Word count and avg. length:
  - some speech patterns are indicative of anger.
- Hashtags:
  - hashtags might be associated with hate speech.
- Exclamation marks:
  - can be an indicator of rage.
- Question marks:
  - people often use rhetorical questions to show anger.
- Uppercase usage:
  - can be an indicator of anger.
- Sentiment:
  - there might be a link between use of negative words and hate speech.



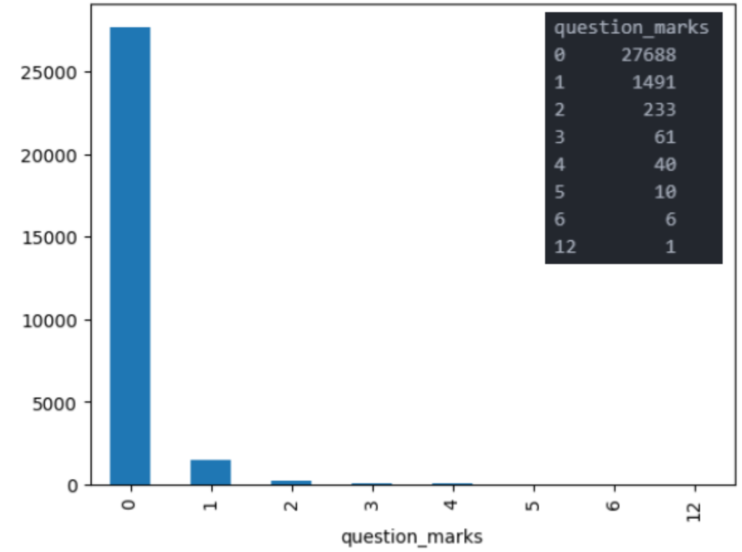
	tweet	label	sentiment
id			
24643	mad @user interracial couple tweet, go fuck yo...	1	-0.5125
22452	@user well there's surprise.... wonder much bi...	1	0.2000
22720	iâ□□ve id checked police zero times 8 months. ...	1	0.0000
21965	hope guy say come trump gets speak mind ð□□.....	1	0.0000
31961	@user #sikh #temple vandalised #calgary, #wso ...	1	0.0000
...	...	...	...
31957	fishing tomorrow @user carnt wait first time 2...	0	0.2500
31958	ate @user isz youuu?ð□□□ð□□□ð□□□ð□□□ð□□□ð□□□ð□...	0	0.0000
31959	see nina turner airwaves trying wrap mantle ge...	0	0.4000
31960	listening sad songs monday morning otw work sad	0	-0.5000
31962	thank @user follow	0	0.0000



label	tweet	word_count	avg_word	hashtags	exclamation_marks	question_marks	upper
1	mad @user interracial couple tweet, go fuck yo...	17	6.438	3	0	0	0
1	@user well there's surprise.... wonder much bi...	14	5.000	0	0	0	0
1	iâ□□ve id checked police zero times 8 months. ...	25	4.042	0	0	0	0
1	hope guy say come trump gets speak mind ð□□.....	24	4.500	1	1	0	0
1	@user #sikh #temple vandalised #calgary, #wso ...	13	5.500	4	0	0	0
...	...	...	...	...	...	...	...
0	fishing tomorrow @user carnt wait first time 2...	13	4.455	0	0	0	0
0	ate @user isz youuu?ð□□□ð□□□ð□□□ð□□□ð□□□ð□□□ð□...	6	12.600	0	0	1	0
0	see nina turner airwaves trying wrap mantle ge...	25	4.652	2	0	0	0
0	listening sad songs monday morning otw work sad	15	3.769	0	0	0	0
0	thank @user follow	8	4.167	0	0	0	0

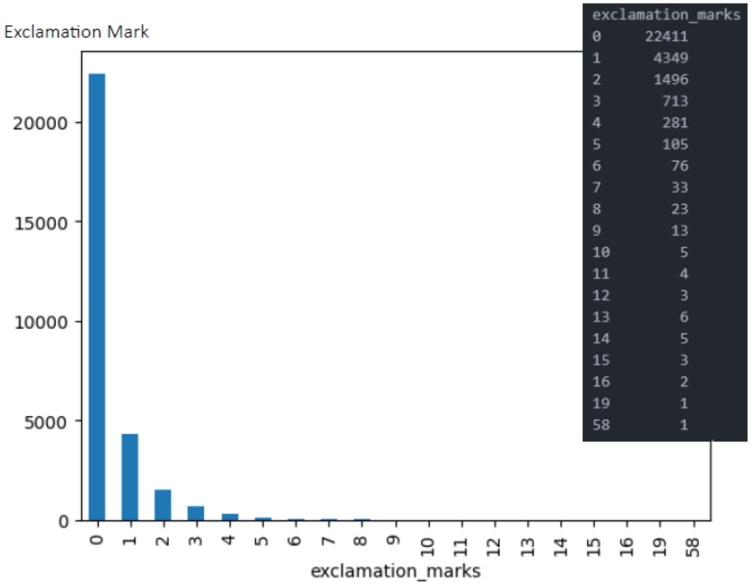
Extra Training Features

Question Marks



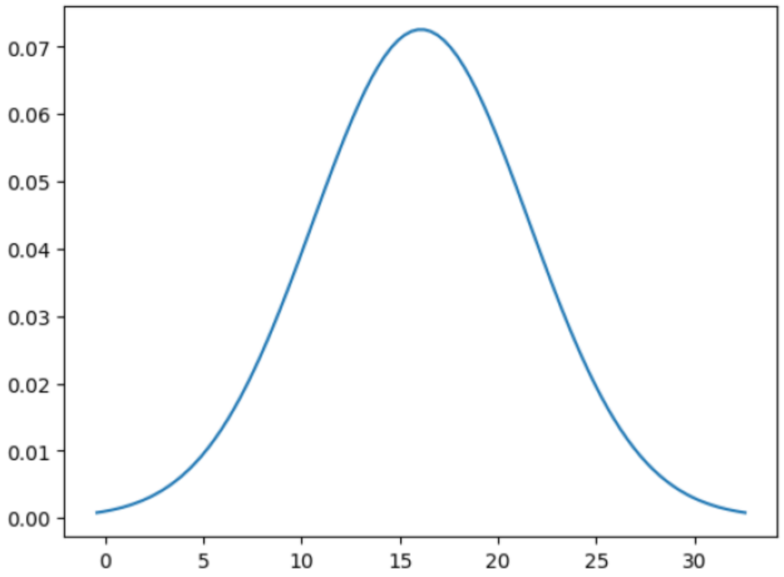
```
question_marks
0       27688
1        1491
2         233
3          61
4          40
5          10
6           6
12          1
```

Exclamation Mark



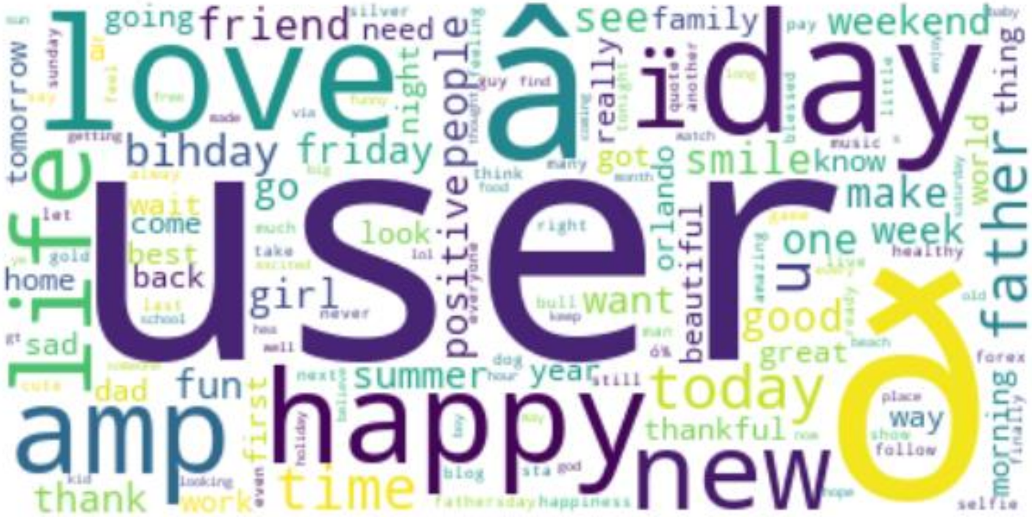
```
exclamation_marks
0       22411
1        4349
2        1496
3         713
4         281
5         105
6          76
7          33
8          23
9          13
10          5
11          4
12          3
13          6
14          5
15          3
16          2
19          1
58          1
```

Word count of tweets

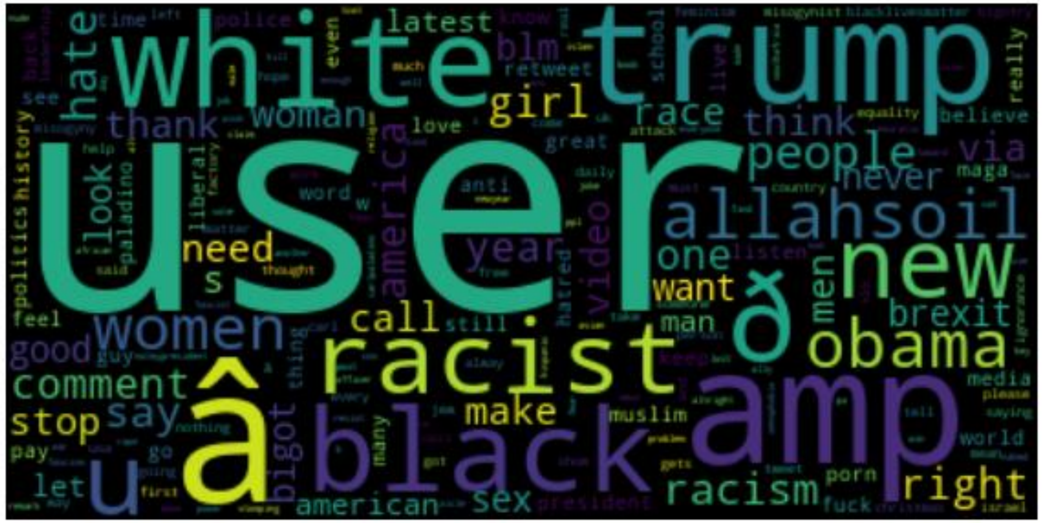


Textcloud of Common Words

Non-Hate Comments



Hate Comments

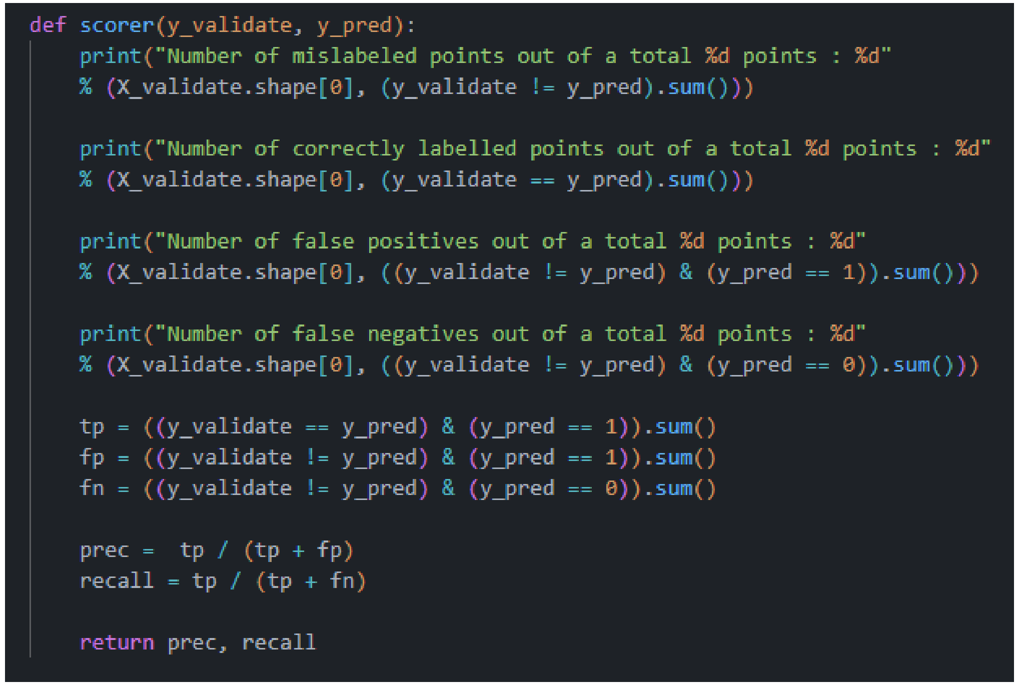


Models Foreword

We must choose whether we wish to minimizes false negatives (stricter model) or false positives more (more lenient model).

- Users don’t want their content to be flagged when it’s not hate speech.
- Advertisers don’t want hate speech at all; if we miss it then this erodes their confidence in us.
- Users can flag missed hate speech and it can be manually reviewed

This must be decided by stakeholders, so we primarily use F1-score, precision, and recall.



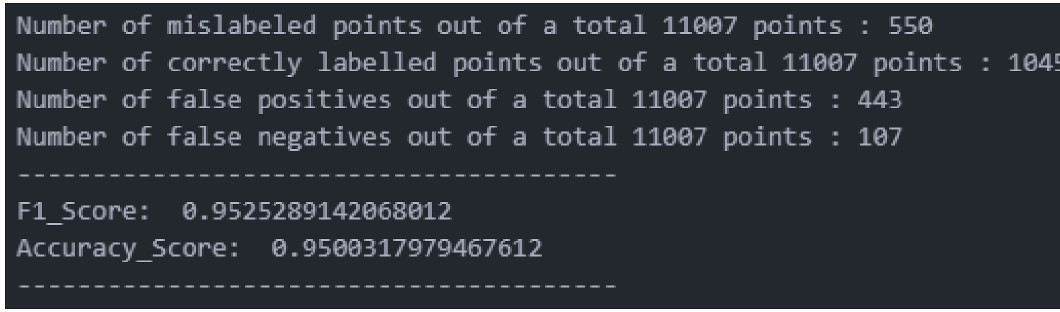
```
def scorer(y_validate, y_pred):
    print("Number of mislabeled points out of a total %d points : %d"
    % (X_validate.shape[0], (y_validate != y_pred).sum()))

    print("Number of correctly labelled points out of a total %d points : %d"
    % (X_validate.shape[0], (y_validate == y_pred).sum()))

    print("Number of false positives out of a total %d points : %d"
    % (X_validate.shape[0], ((y_validate != y_pred) & (y_pred == 1)).sum()))

    print("Number of false negatives out of a total %d points : %d"
    % (X_validate.shape[0], ((y_validate != y_pred) & (y_pred == 0)).sum()))

    tp = ((y_validate == y_pred) & (y_pred == 1)).sum()
    fp = ((y_validate != y_pred) & (y_pred == 1)).sum()
    fn = ((y_validate != y_pred) & (y_pred == 0)).sum()

    prec =  tp / (tp + fp)
    recall = tp / (tp + fn)

    return prec, recall
```

K-Nearest Neighbor (KNN)

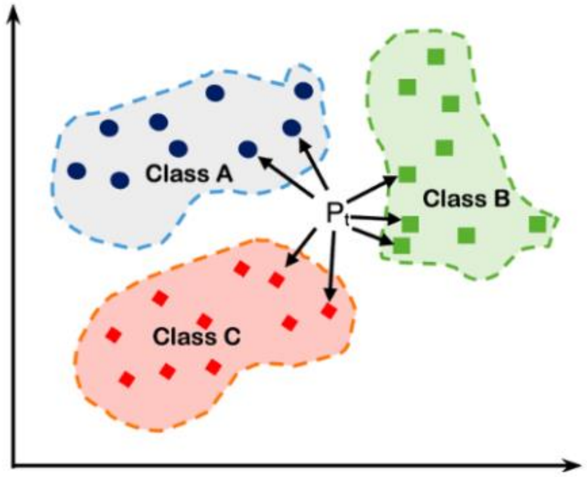
Hyperparameters

- n_neighbours=1: any higher n was reducing the F-1 score.
- weights=‘distance’: can be uniform or distance. Using distance makes weights inversely proportional to the distance.
- leaf_size=30: default
- p=1: uses l1 norm (max) for distance rather than Euclidian distance.

Model can’t be graphed since it uses 7 features plus the TF-IDF transformer.



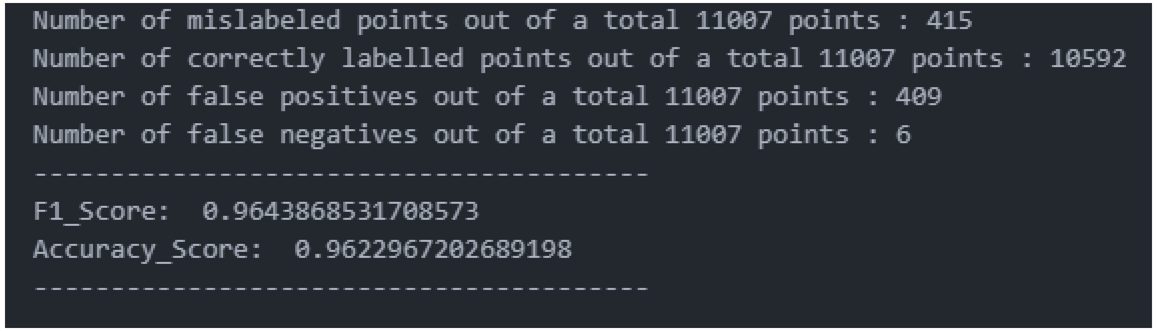
```
Number of mislabeled points out of a total 11007 points : 550
Number of correctly labelled points out of a total 11007 points : 1045
Number of false positives out of a total 11007 points : 443
Number of false negatives out of a total 11007 points : 107
----------------------------------------
F1_Score:  0.9525289142068012
Accuracy_Score:  0.9500317979467612
----------------------------------------
```



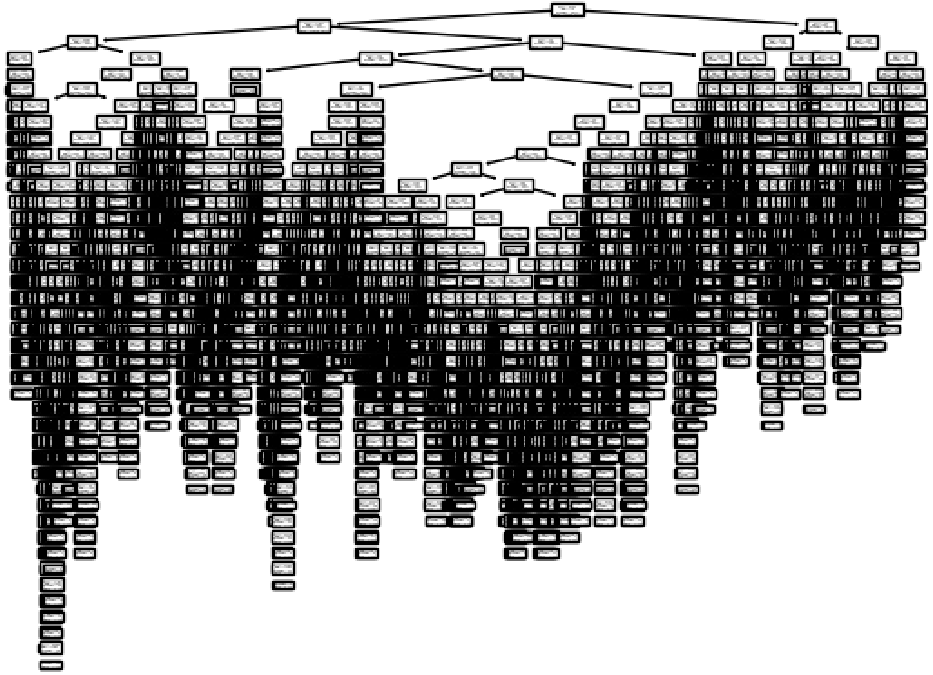
K Nearest Neighbors

Decision Tree

- Best model by every metric.
- Especially high recall.
  - Low number of false negatives.
  - Stricter model
- No max depth specified so the tree is quite complex.
- gini was the best criterion.
  - log_loss and entropy were also tried.

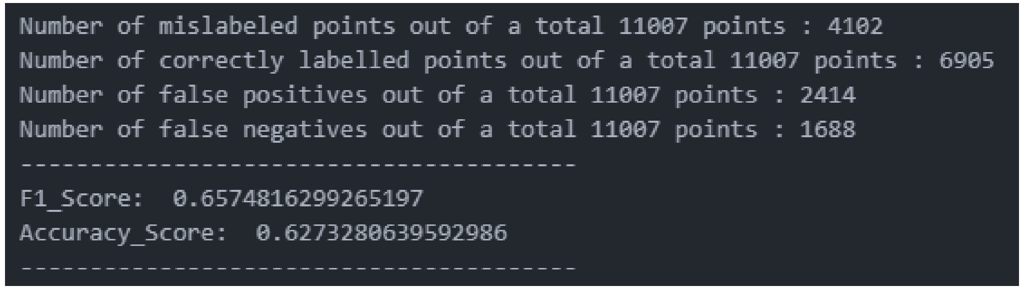


```
Number of mislabeled points out of a total 11007 points : 415
Number of correctly labelled points out of a total 11007 points : 10592
Number of false positives out of a total 11007 points : 409
Number of false negatives out of a total 11007 points : 6
----------------------------------------
F1_Score:  0.9643868531708573
Accuracy_Score:  0.9622967202689198
----------------------------------------
```

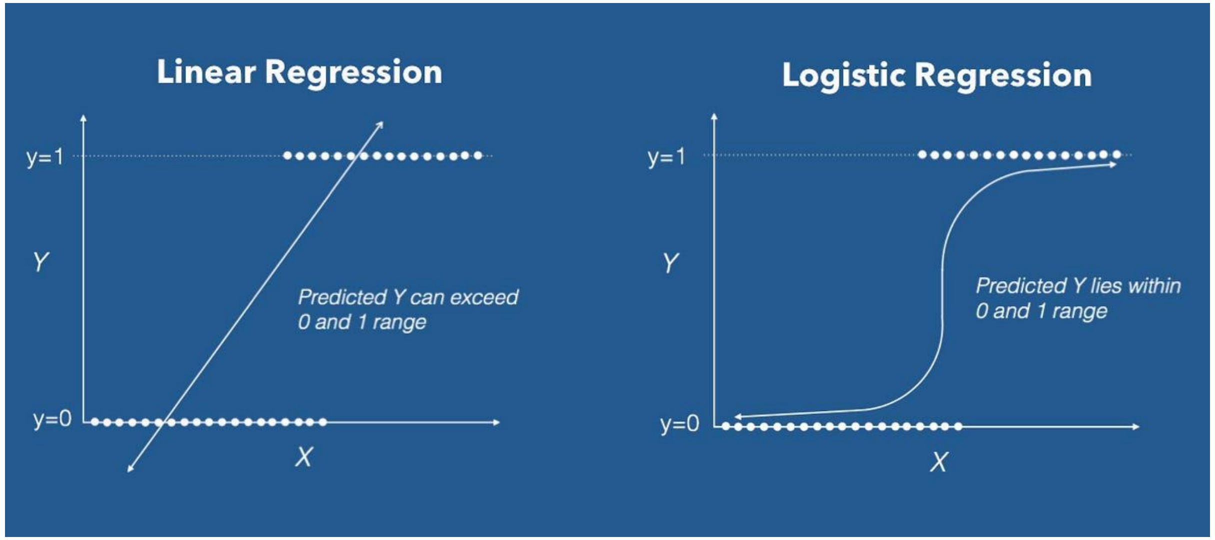


Logistic Regression

- Bad model overall.
- Could have benefitted from better hyper tuning of parameters.
  - Greater regularization.
  - Change in intercept bias.
- Balanced class weights originally were used because the minority class wasn't up sampled.
  - After up sampling, this makes very little difference.
- Random sampling for all models can cause a bit of imbalance between the two classes (negligible if randomly selected).

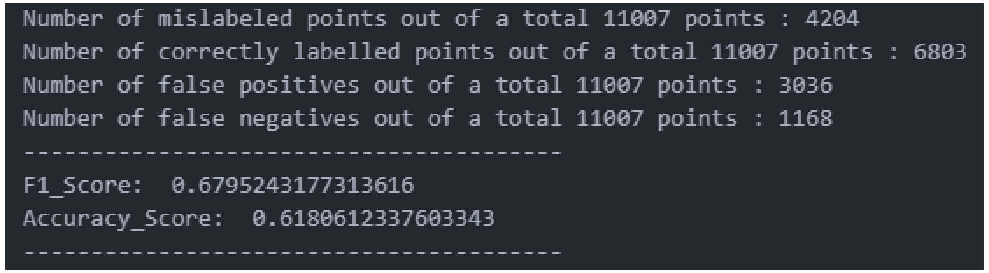


```
Number of mislabeled points out of a total 11007 points : 4102
Number of correctly labelled points out of a total 11007 points : 6905
Number of false positives out of a total 11007 points : 2414
Number of false negatives out of a total 11007 points : 1688
----------------------------------------
F1_Score:  0.6574816299265197
Accuracy_Score:  0.6273280639592986
----------------------------------------
```


Linear Regression
y=1
Y
y=0
X
Predicted Y can exceed 0 and 1 range
Logistic Regression
y=1
Y
y=0
X
Predicted Y lies within 0 and 1 range

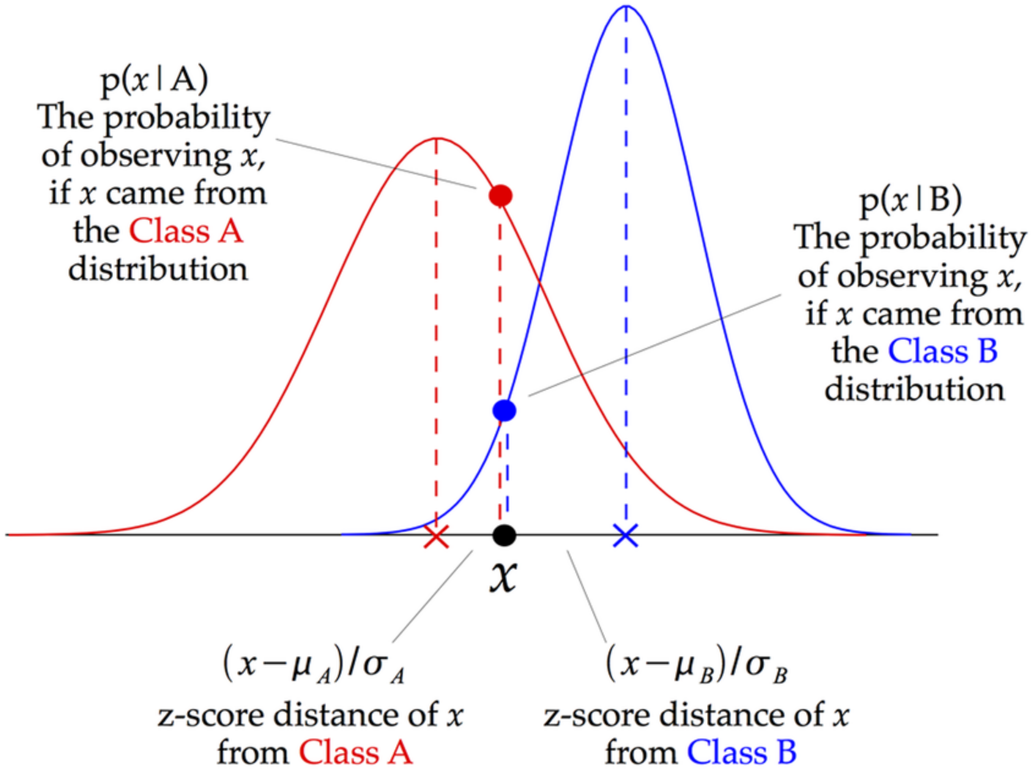
Gaussian Naïve Bayes

- Bad model overall.
- TF-IDF transformation couldn’t be used because it outputs a sparse matrix and GNB needs a dense one.
    - Could be fixed if pipeline not used.
- Could be improved using CountVector transformation if data pre-processing style is changed.



```
Number of mislabeled points out of a total 11007 points : 4204
Number of correctly labelled points out of a total 11007 points : 6803
Number of false positives out of a total 11007 points : 3036
Number of false negatives out of a total 11007 points : 1168
----------------------------------------
F1_Score:  0.6795243177313616
Accuracy_Score:  0.6180612337603343
----------------------------------------
```

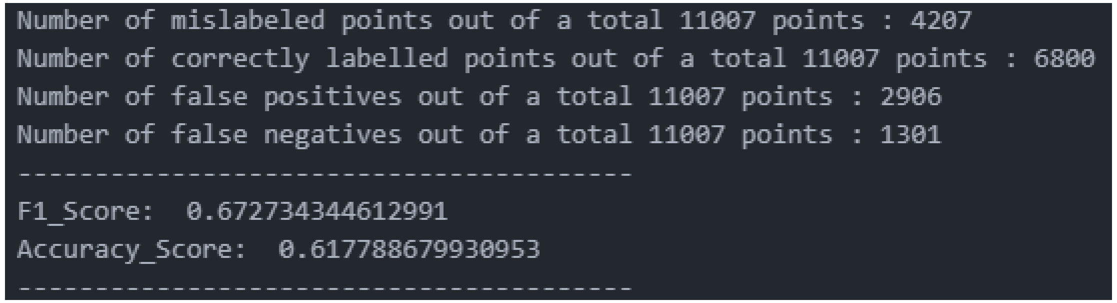
$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$



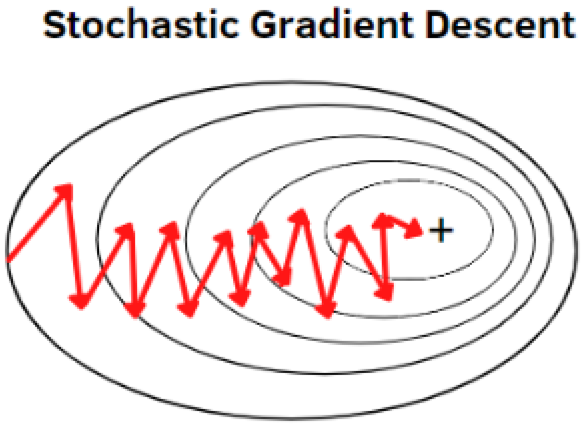
p(x | A)
The probability of observing x, if x came from the Class A distribution
p(x | B)
The probability of observing x, if x came from the Class B distribution
x
(x − μ_A)/σ_A
z-score distance of x from Class A
(x − μ_B)/σ_B
z-score distance of x from Class B

Stochastic Gradient Descent (SGD)

- Bad model overall.
- Could be improved using CountVector transformation if data pre-processing style is changed.
- An accelerated gradient descent might be beneficial.
- Changing the loss function using might be beneficial.

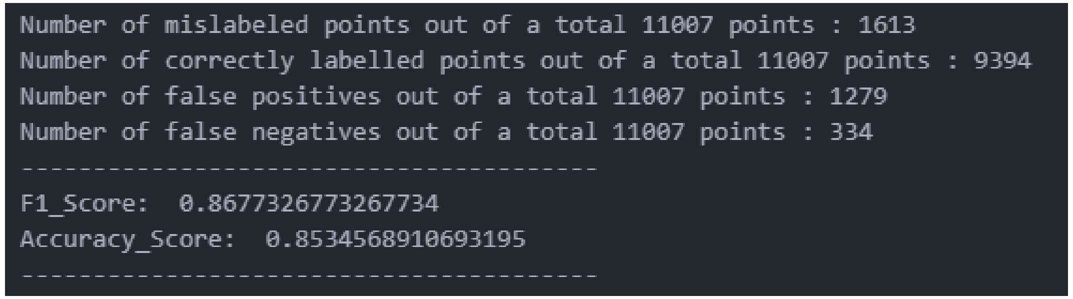


```
Number of mislabeled points out of a total 11007 points : 4207
Number of correctly labelled points out of a total 11007 points : 6800
Number of false positives out of a total 11007 points : 2906
Number of false negatives out of a total 11007 points : 1301
----------------------------------------
F1_Score:  0.672734344612991
Accuracy_Score:  0.617788679930953
----------------------------------------
```

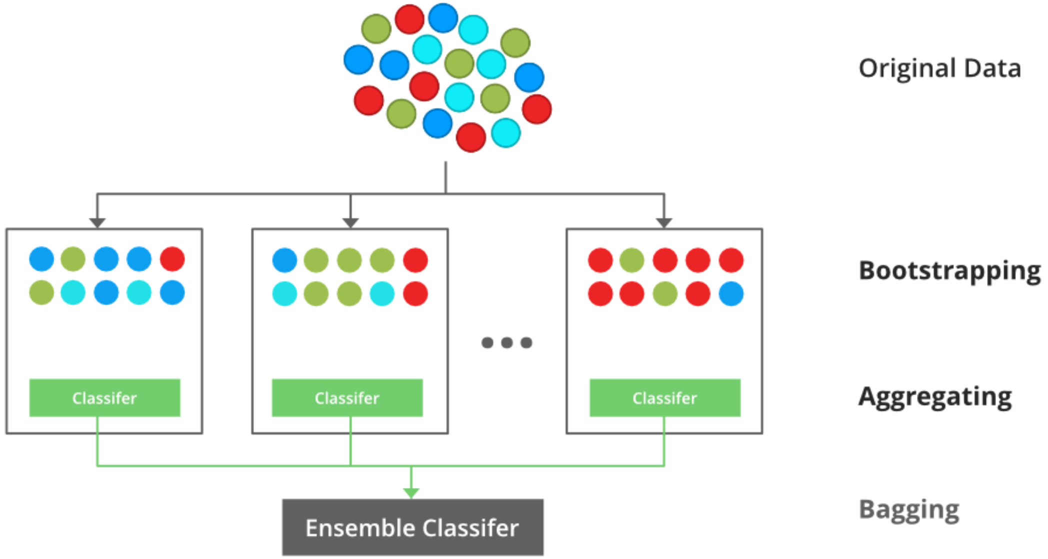


XGB Classifier

- Worked moderately well
- Ensemble classifier
  - Would likely benefit from more features for training.
- A list of commonly used words in hate speech could be used.
  - List would have to be updated regularly since language changes quickly on social media.
  - Could introduce political bias.



```
Number of mislabeled points out of a total 11007 points : 1613
Number of correctly labelled points out of a total 11007 points : 9394
Number of false positives out of a total 11007 points : 1279
Number of false negatives out of a total 11007 points : 334
----------------------------------------
F1_Score:  0.8677326773267734
Accuracy_Score:  0.8534568910693195
----------------------------------------
```

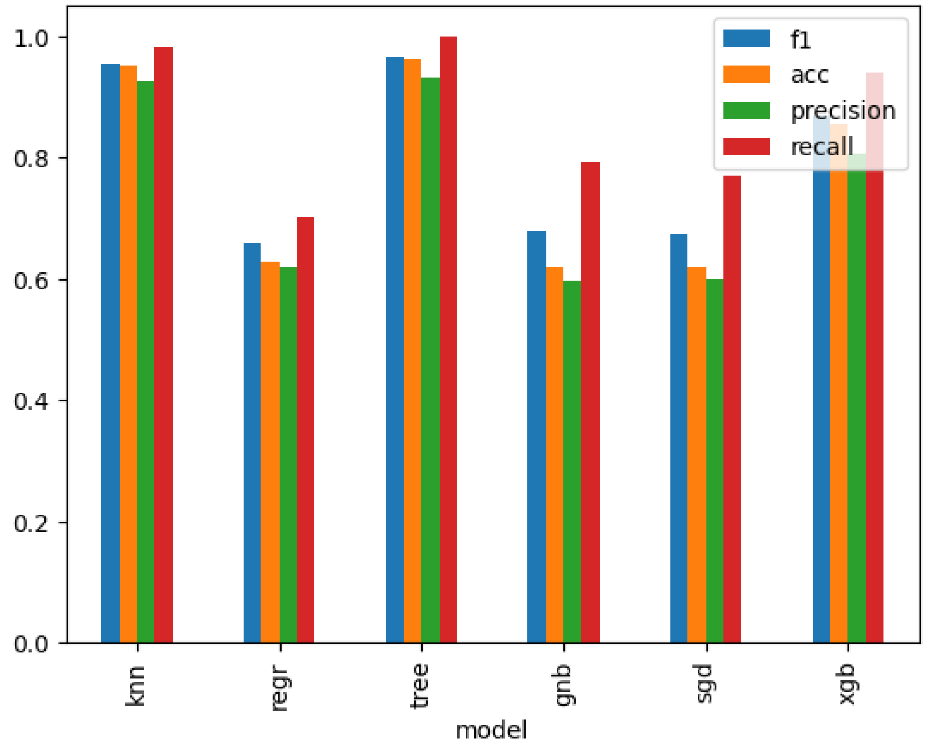


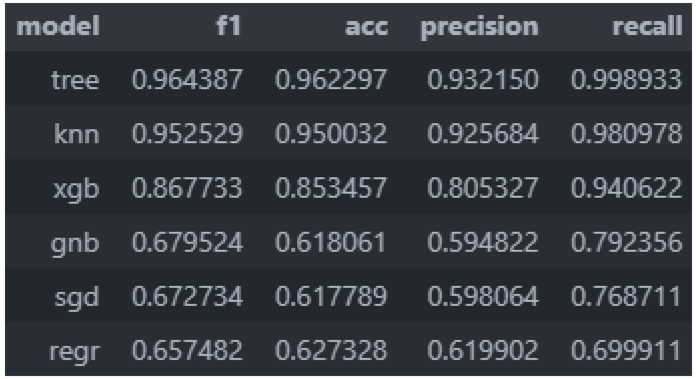
Results

Decision tree performed the best in every category. It was exceptional at recall (i.e., minimizing false negatives).

Why black box models are suitable:

- Don’t want political views to affect what is viewed as hate speech (greater censorship).
- Better than individuals reviewing hate speech since model is consistent.
- Bias can exist from chosen data and how it is labelled, but not after the model is trained.
- However, model will have to be retrained to keep up with changing language.





model	f1	acc	precision	recall
tree	0.964387	0.962297	0.932150	0.998933
knn	0.952529	0.950032	0.925684	0.980978
xgb	0.867733	0.853457	0.805327	0.940622
gnb	0.679524	0.618061	0.594822	0.792356
sgd	0.672734	0.617789	0.598064	0.768711
regr	0.657482	0.627328	0.619902	0.699911

Thank You

Data Glacier

Your Deep Learning Partner