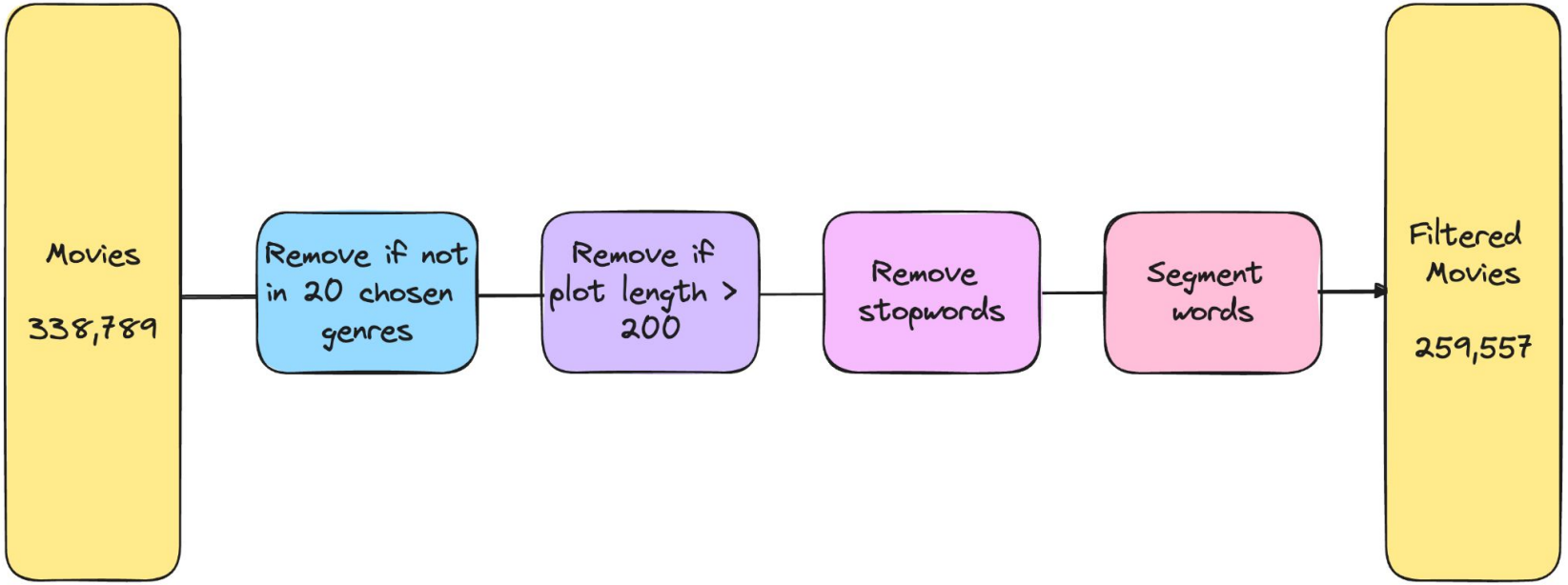
Predicting Movie Genres Based on Plot Summaries

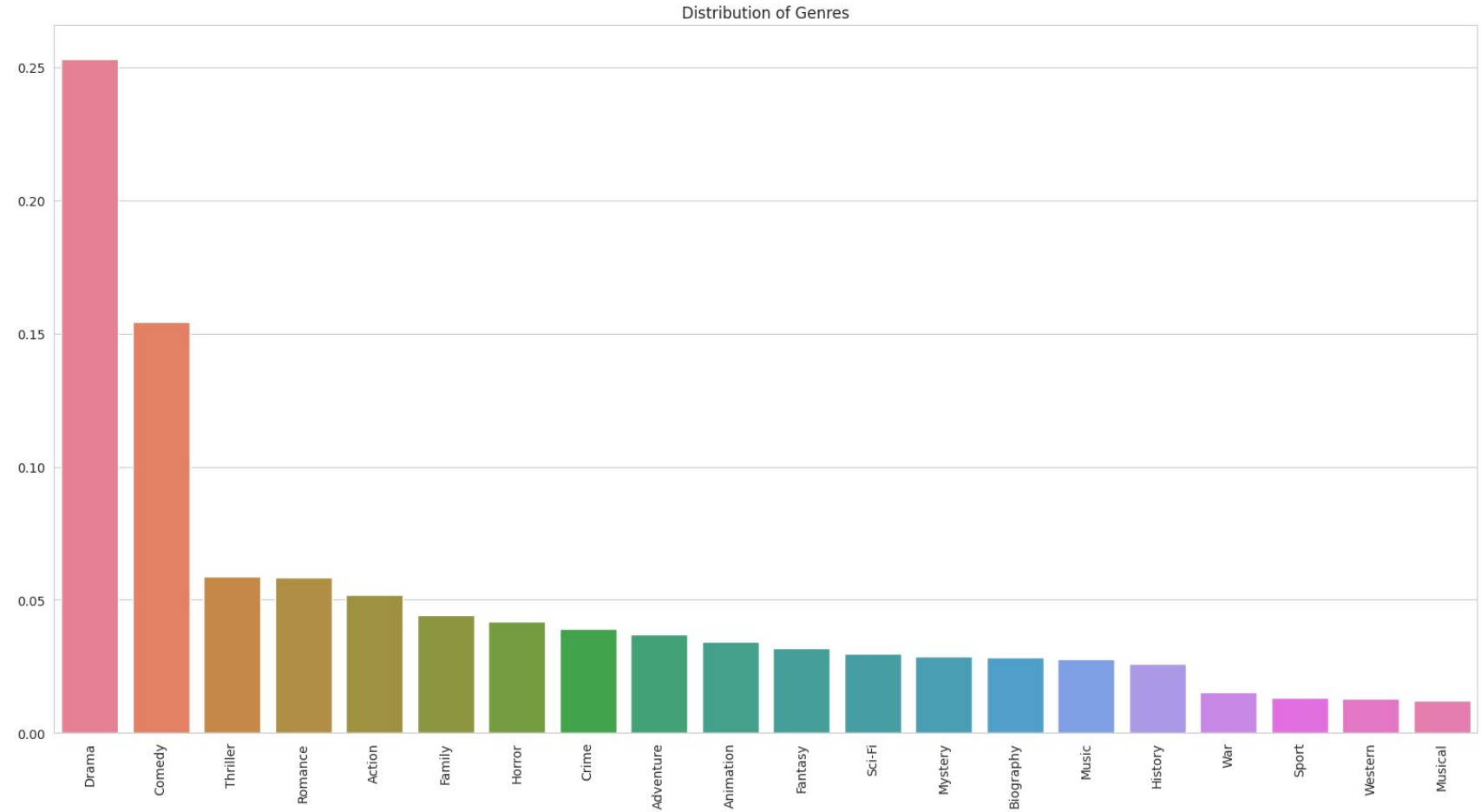
Team-13: AegiFi
Akshit Sinha
Adyansh Kakran

Contents

Data preprocessing



Exploratory Data Analysis

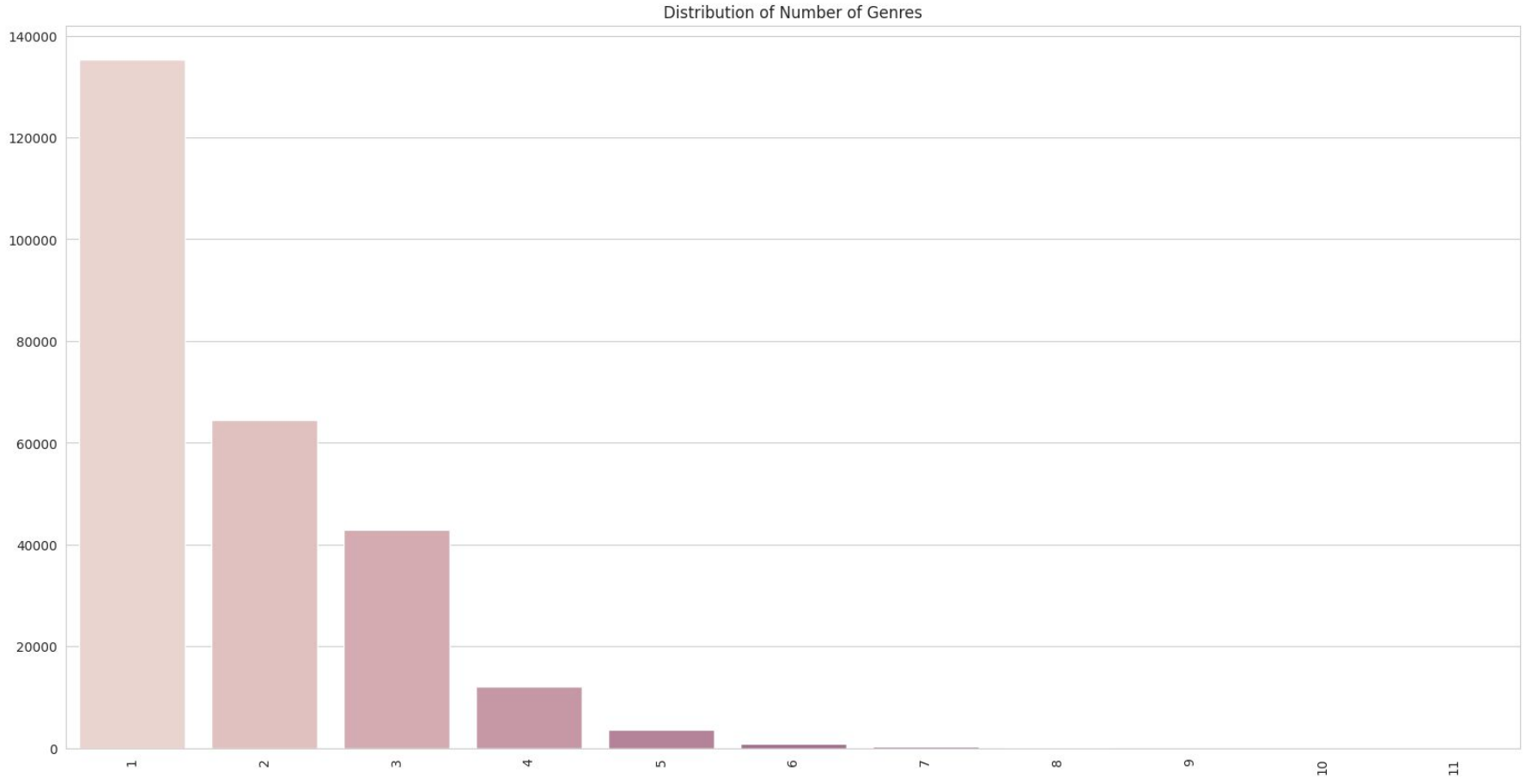

Distribution of Genres
0.25
0.20
0.15
0.10
0.05
0.00
Drama
Comedy
Thriller
Romance
Action
Family
Horror
Crime
Adventure
Animation
Fantasy
Sci-Fi
Mystery
Biography
Music
History
War
Sport
Western
Musical

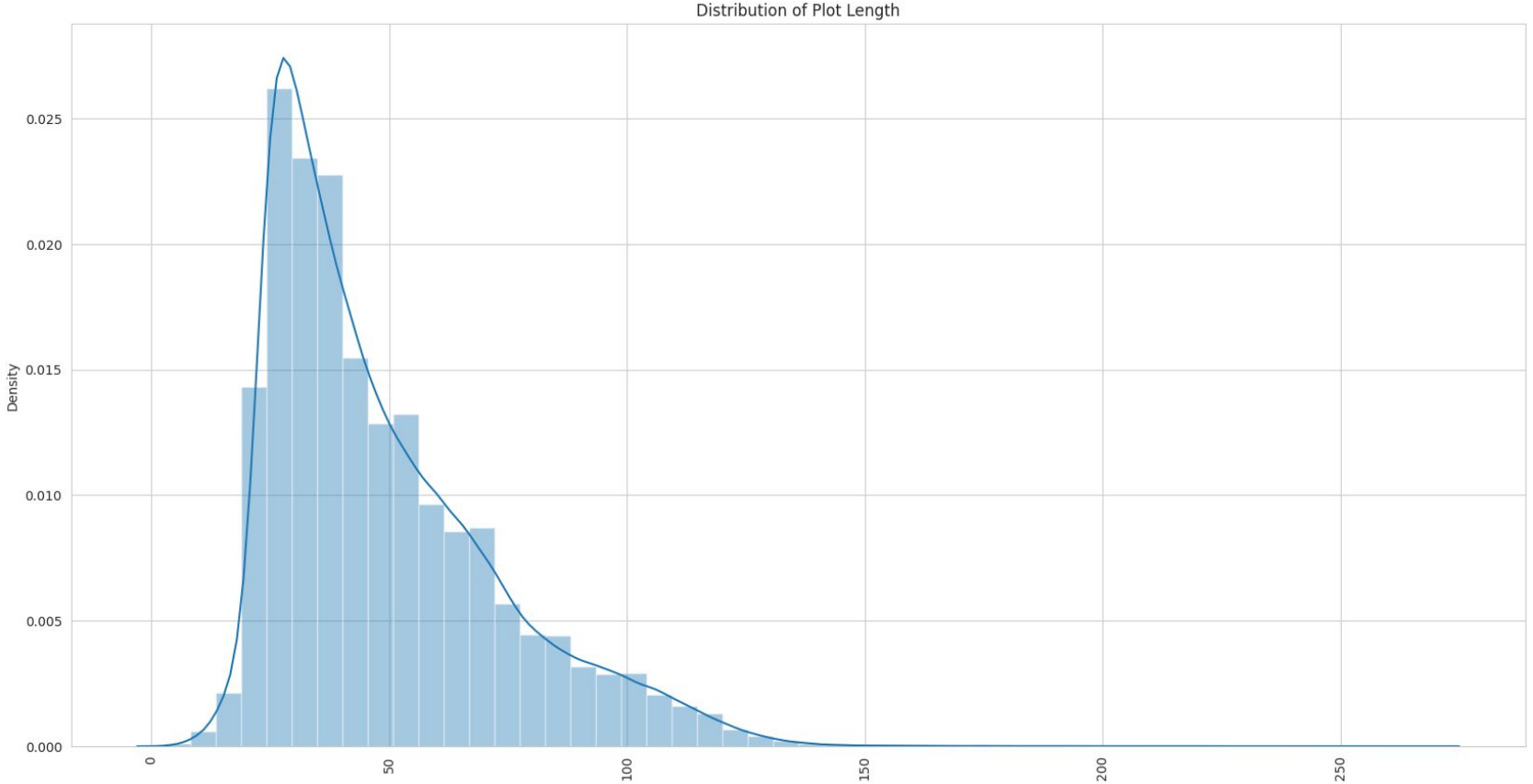
Word Cloud

Observations:

Common words usually generic

Some genre specific words: love, war, couple, etc


Distribution of Number of Genres
140000
120000
100000
80000
60000
40000
20000
0
1
2
3
4
5
6
7
8
9
10
11


Distribution of Plot Length
Density
0.025
0.020
0.015
0.010
0.005
0.000
0
50
100
150
200
250

Machine Learning Models

Naive Bayes

Two approaches used:

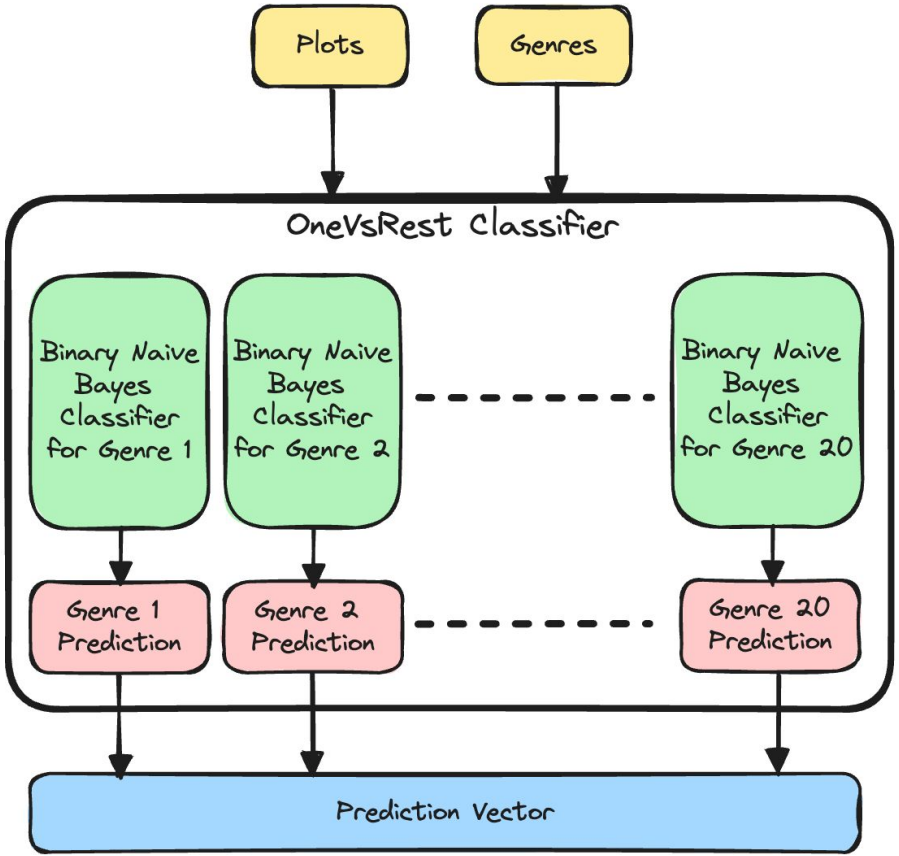
Bernoulli Naive Bayes using OneVsRest method

- Construct 20 individual Binary Naive Bayes classifiers for each genre.

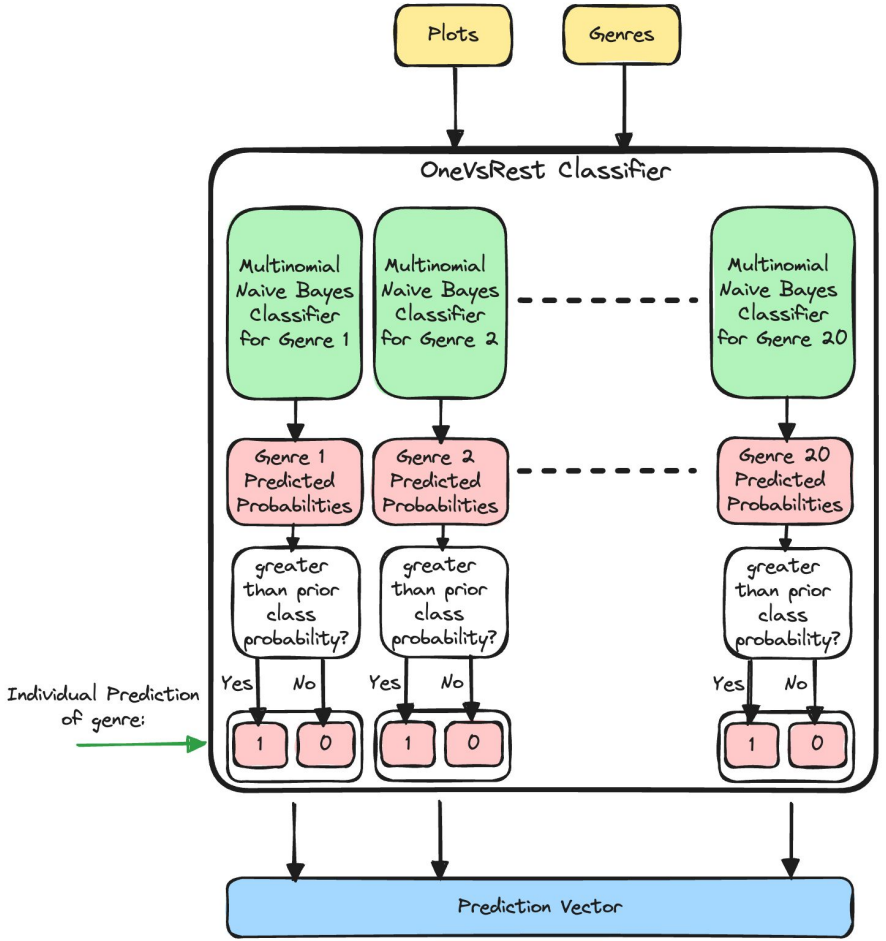
Multinomial Naive Bayes

- Predict a genre if posterior probability is greater than prior probability

Binary Naive Bayes



Multinomial Naive Bayes



XGBoost

Get pre trained word embeddings using the Google News Word2Vec corpus

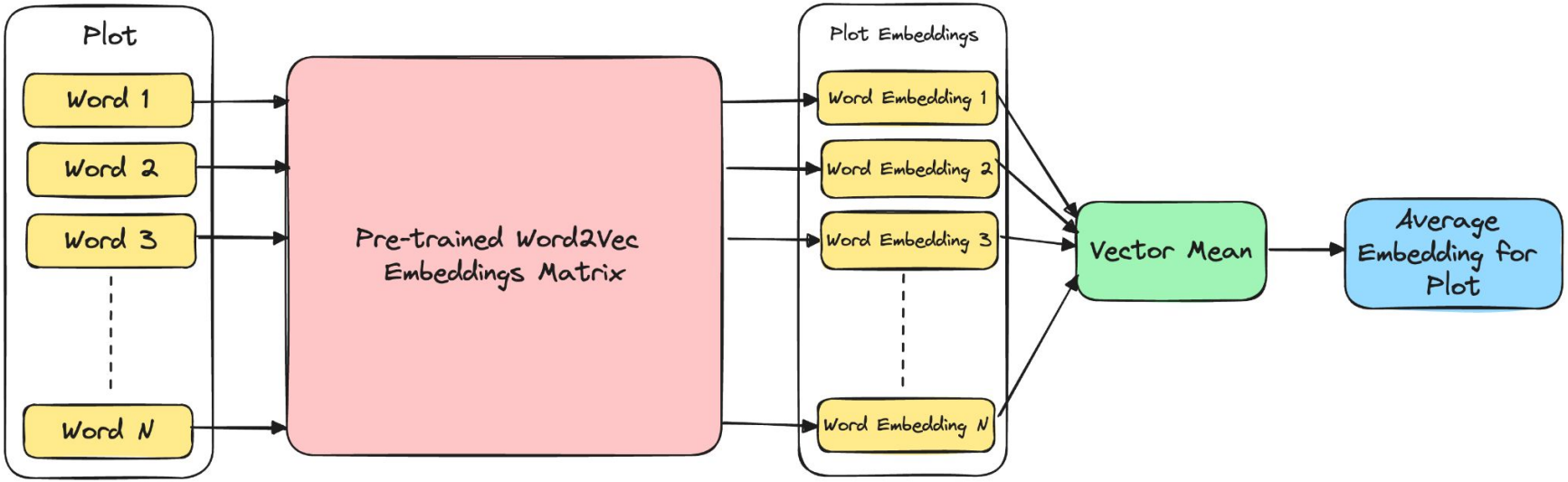
Train an XGBClassifier on these embeddings. Get the output probabilities for the genres

Calculate the threshold probability using the relation:

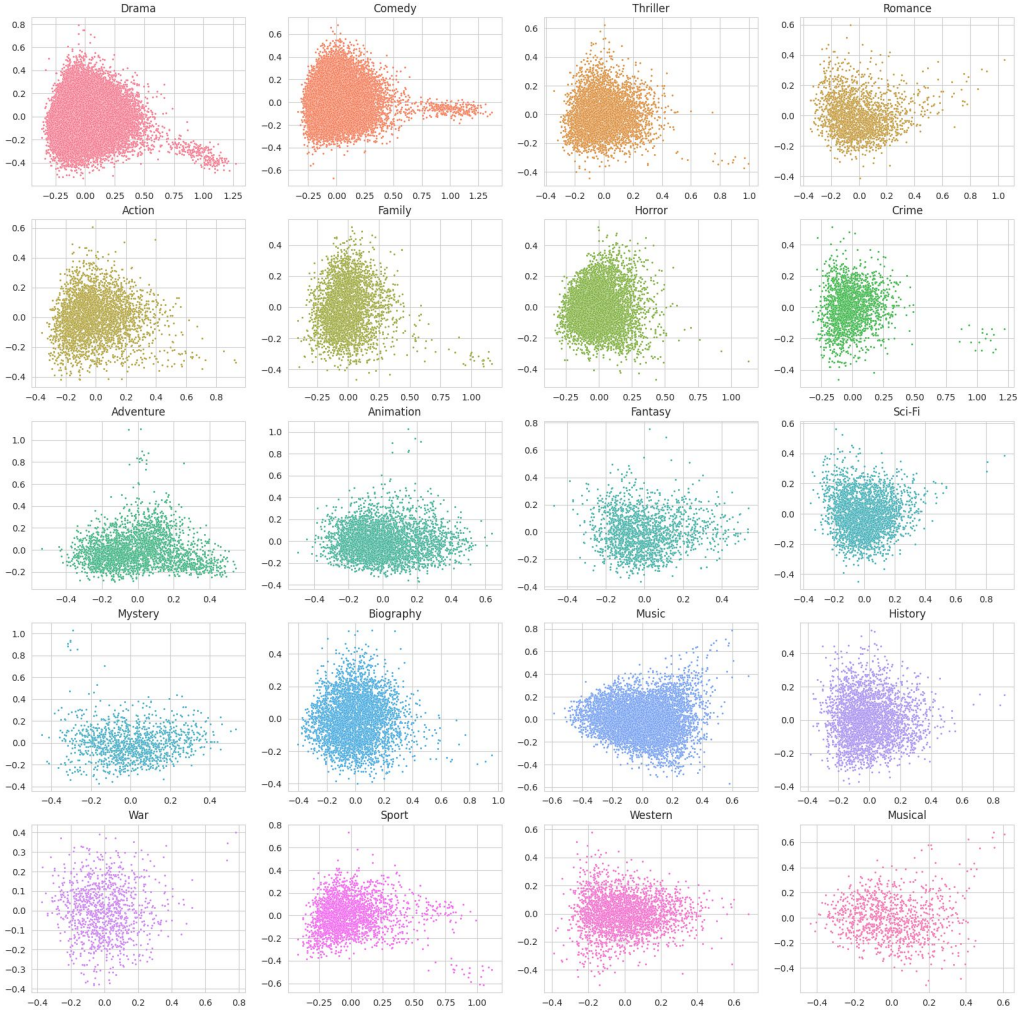
$$t(\mathbf{x}) = \arg\min_{t} \left| \{c \in C\} \ s.t. \ P(c|\mathbf{x}) \leq t \right| + \left| \{c \in (\mathcal{Y} \setminus C)\} \ s.t. \ P(c|\mathbf{x}) \geq t \right|$$

Feed the output probabilities as input and the calculated threshold as the label to an XGBRegressor to learn the best threshold values for the probabilities

Word2Vec Embeddings

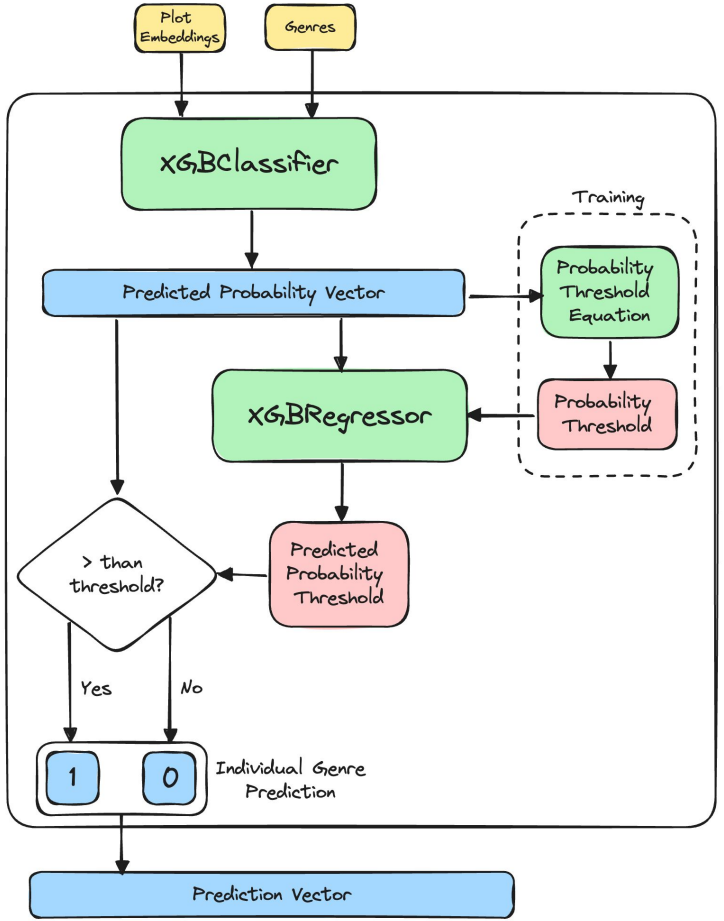


Word2Vec Embeddings



XGBoost Model

$$t(\mathbf{x}) = \arg\min_{t} \left| \{c \in C\} \ s.t. \ P(c|\mathbf{x}) \leq t \right| + \left| \{c \in (\mathcal{Y} \setminus C)\} \ s.t. \ P(c|\mathbf{x}) \geq t \right|$$



GRU

3 approaches used

Binary GRU:

Output linear layer is fed to a sigmoid activation function, and genres with a sigmoid value > 0.5 are predicted

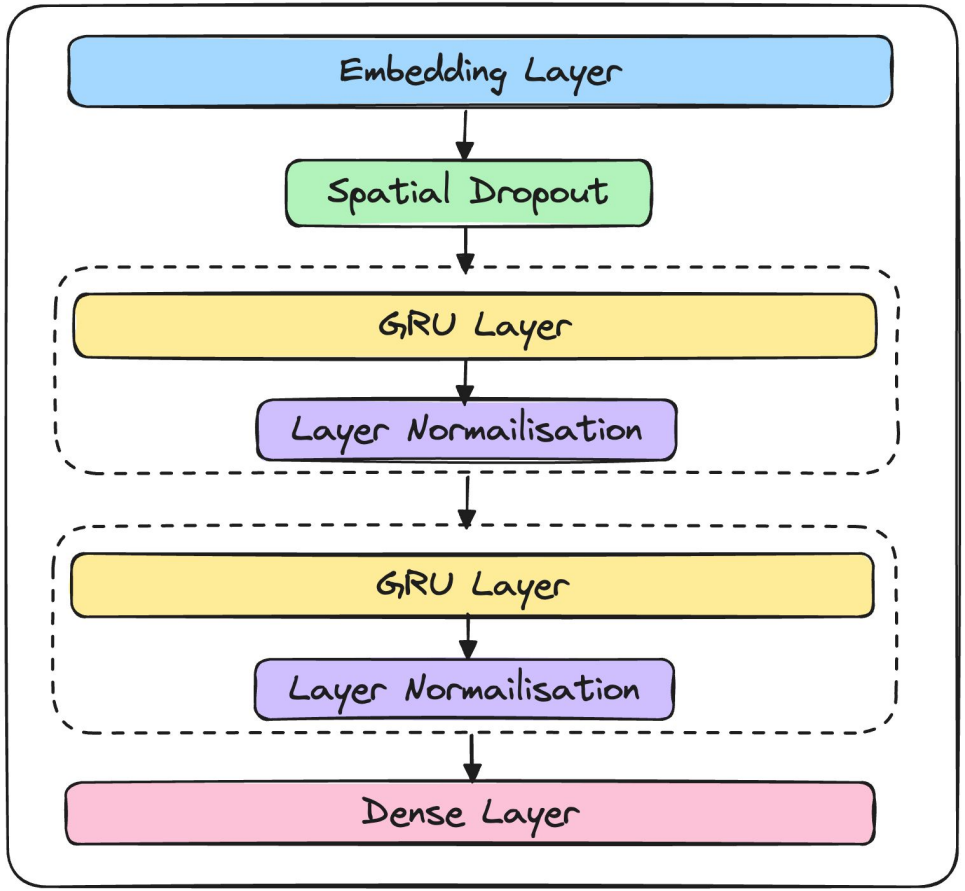
Rank GRU:

Linear layer outputs are interpreted as ranks of the genres. A sigmoid function is applied to convert them to values between 0 and 1, and best probability threshold is calculated as previously, and an XGBRegressor is trained on these values

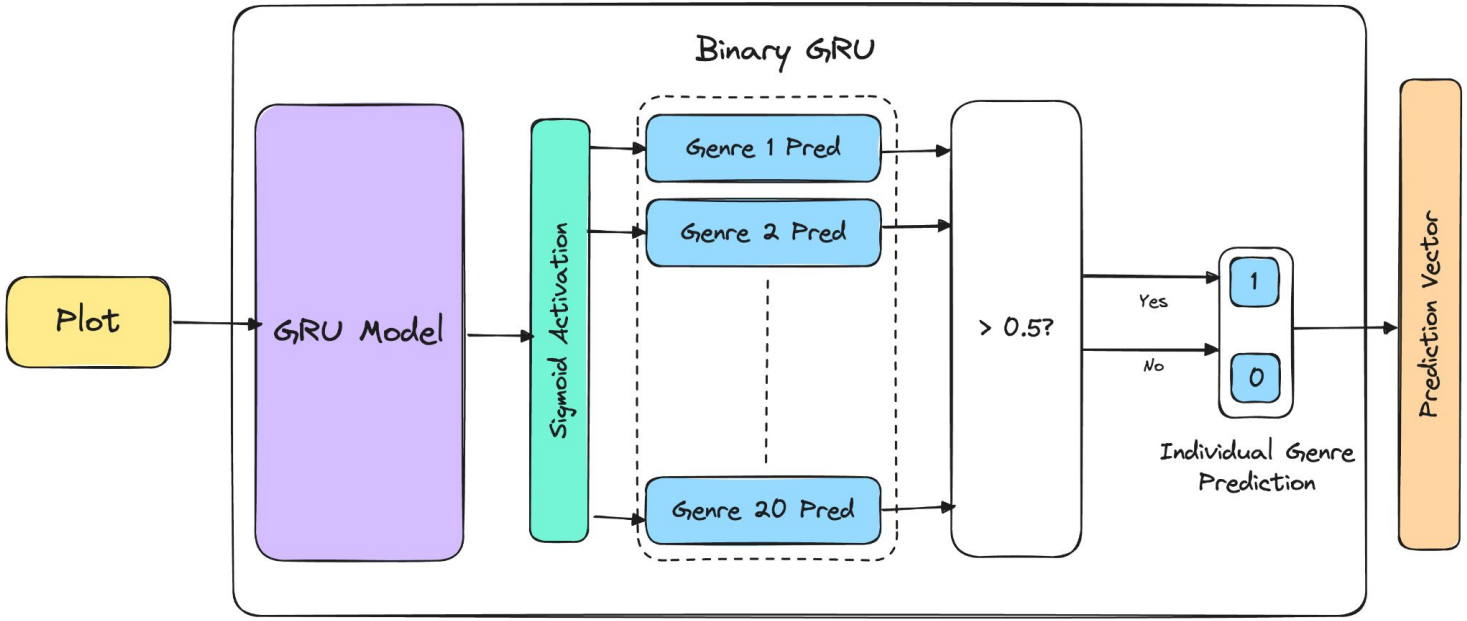
Multinomial GRU:

Outputs are fed into a softmax function to estimate probability of each genre. Further, the target probabilities of each genre are set to 1/k if C is in the set of k correct genres, and 0 otherwise. Network is trained to minimise cross entropy between predicted and target categorical distribution. Again, threshold values are estimated using the previous method.

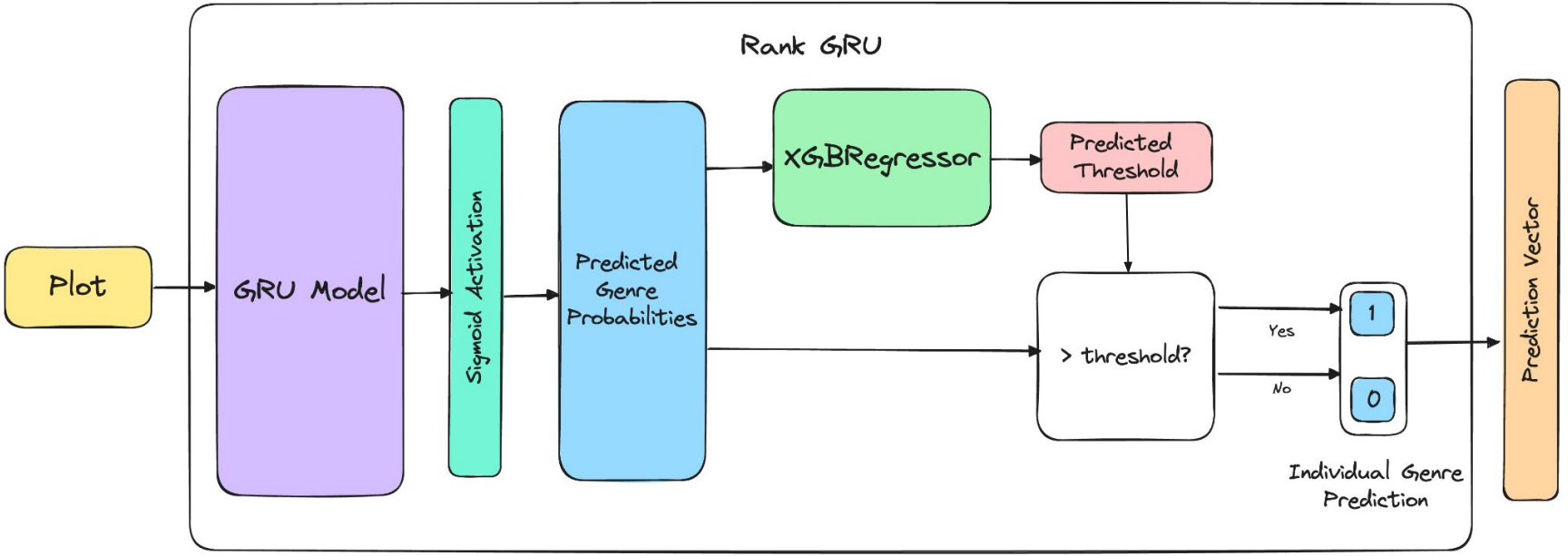
GRU Architecture



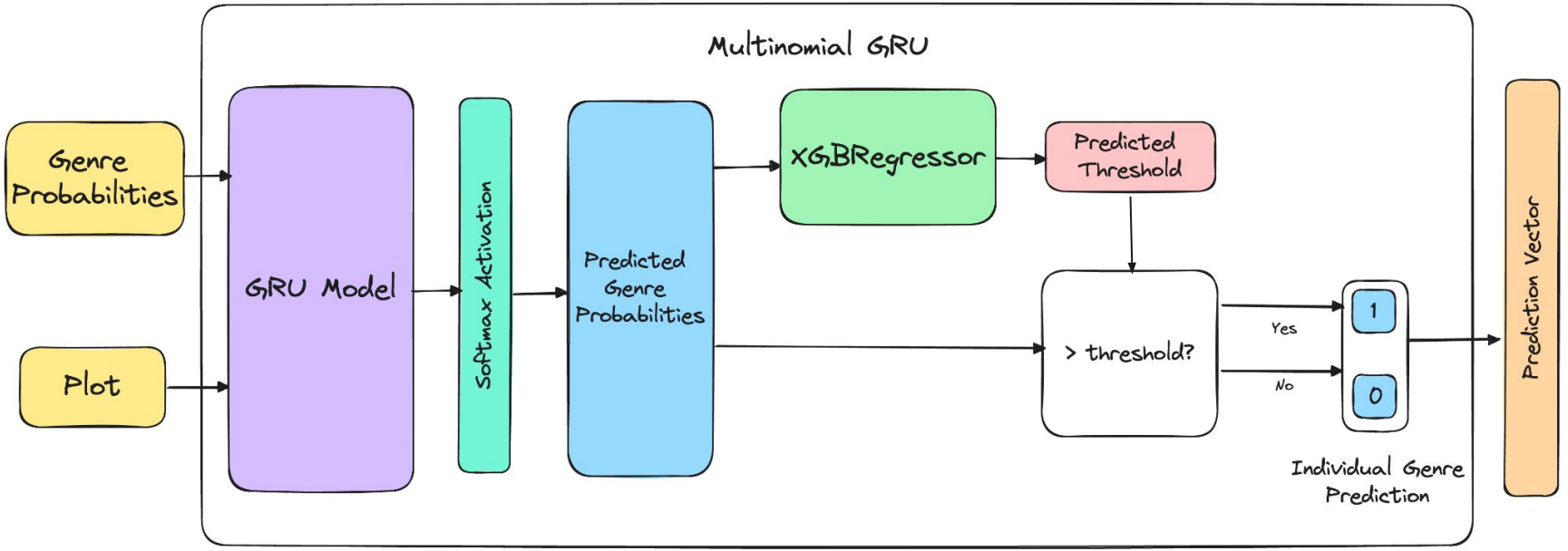
Binary GRU


Binary GRU
Plot
GRU Model
Sigmoid Activation
Genre 1 Pred
Genre 2 Pred
Genre 20 Pred
> 0.5?
Yes
No
1
0
Individual Genre Prediction
Prediction Vector

Rank GRU


Rank GRU
Plot
GRU Model
Sigmoid Activation
Predicted Genre Probabilities
XGBRegressor
Predicted Threshold
> threshold?
Yes
No
1
0
Individual Genre Prediction
Prediction Vector

Multinomial GRU

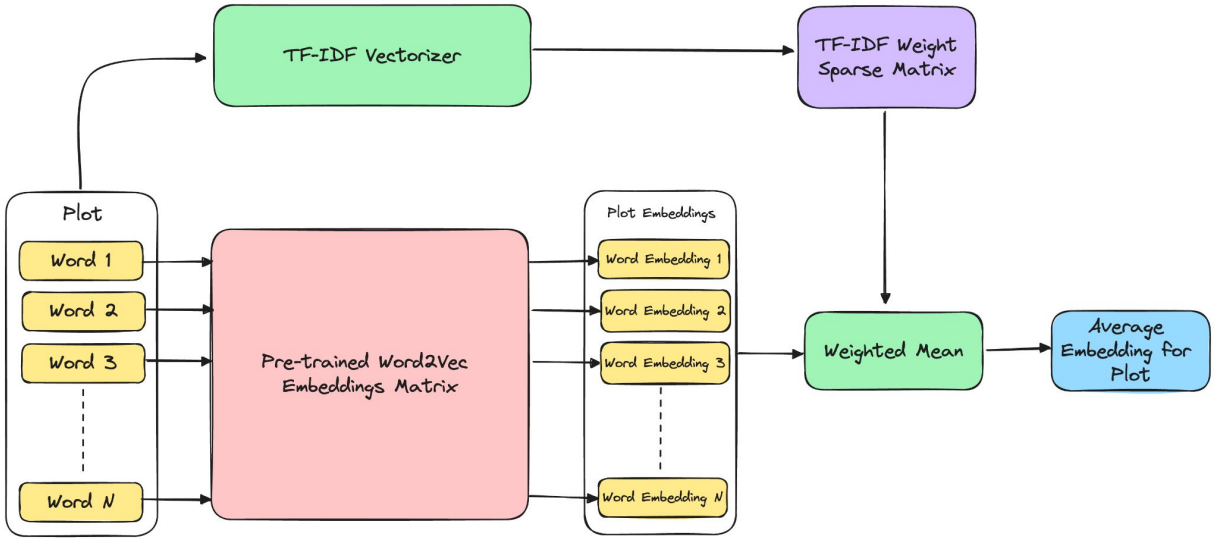


Improvements over Publication

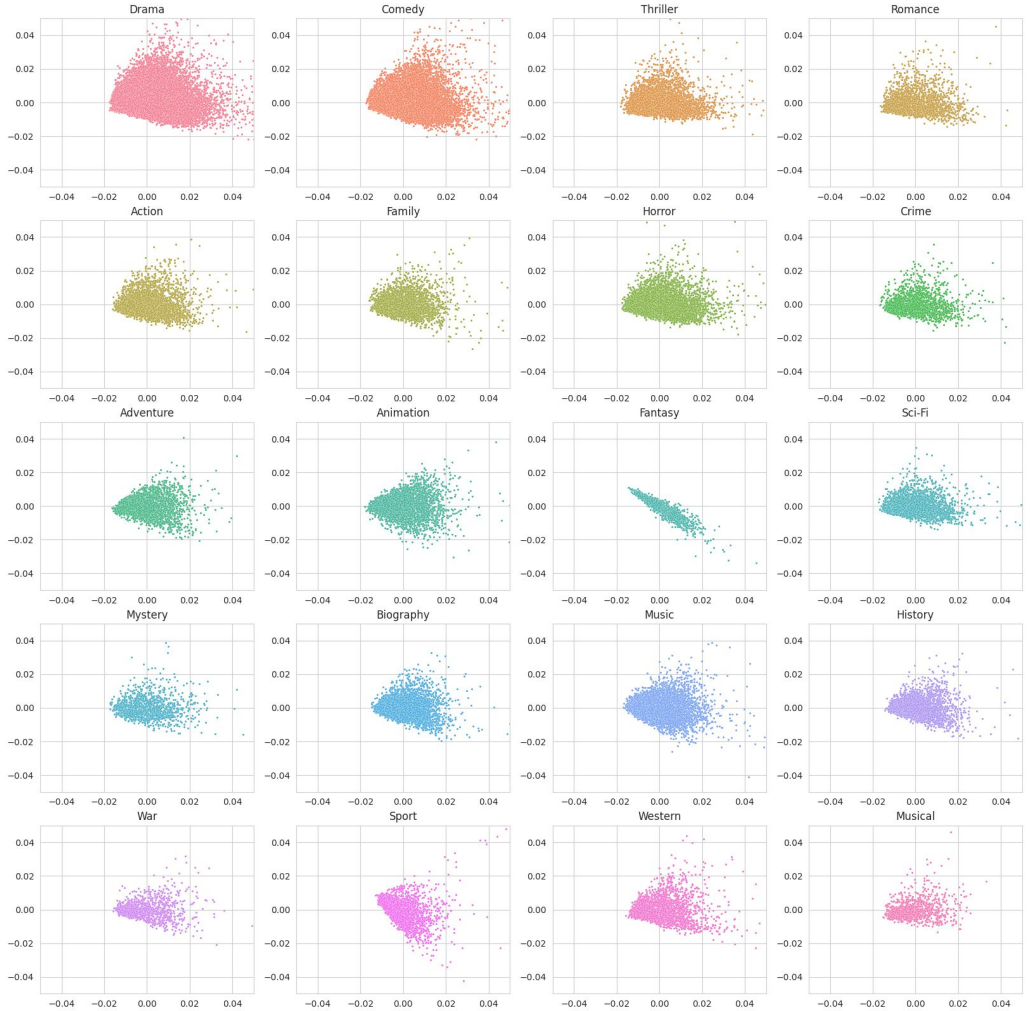
TF-IDF Weighted Word2Vec Embeddings

Problem with word2vec embeddings: same embeddings for UNK

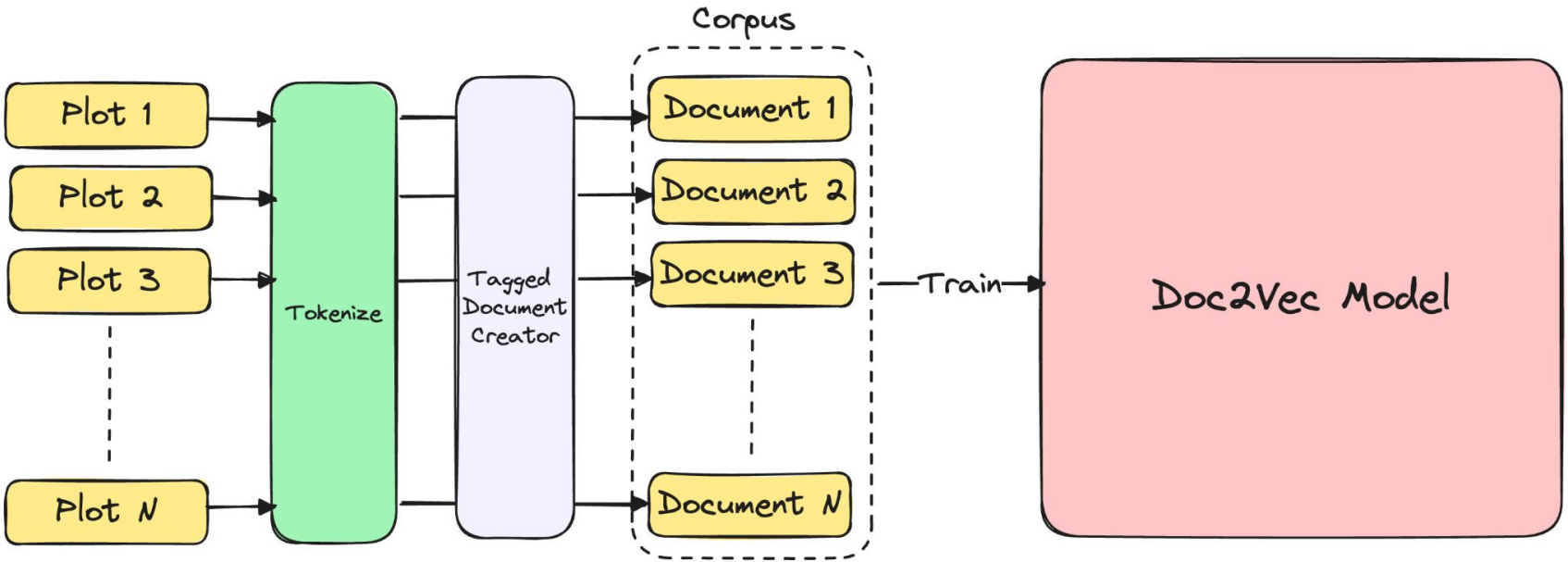
Solved by TF-IDF weighted: a better representation of those UNK tokens



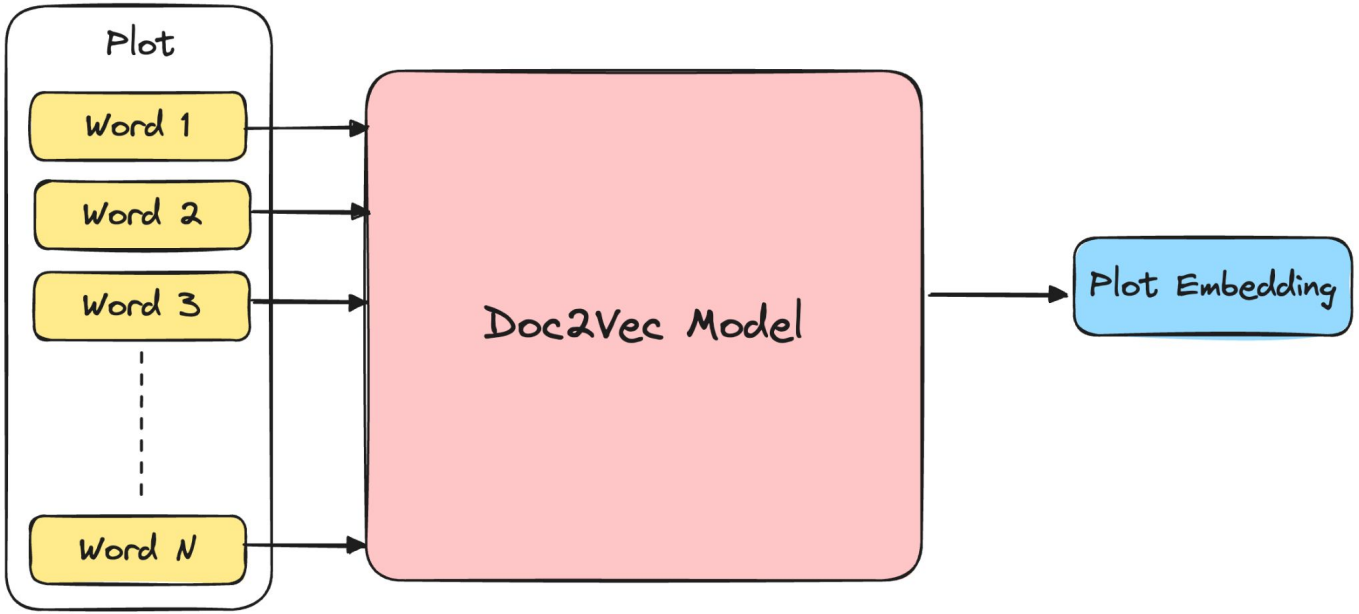
TF-IDF Weighted Word2Vec Embeddings



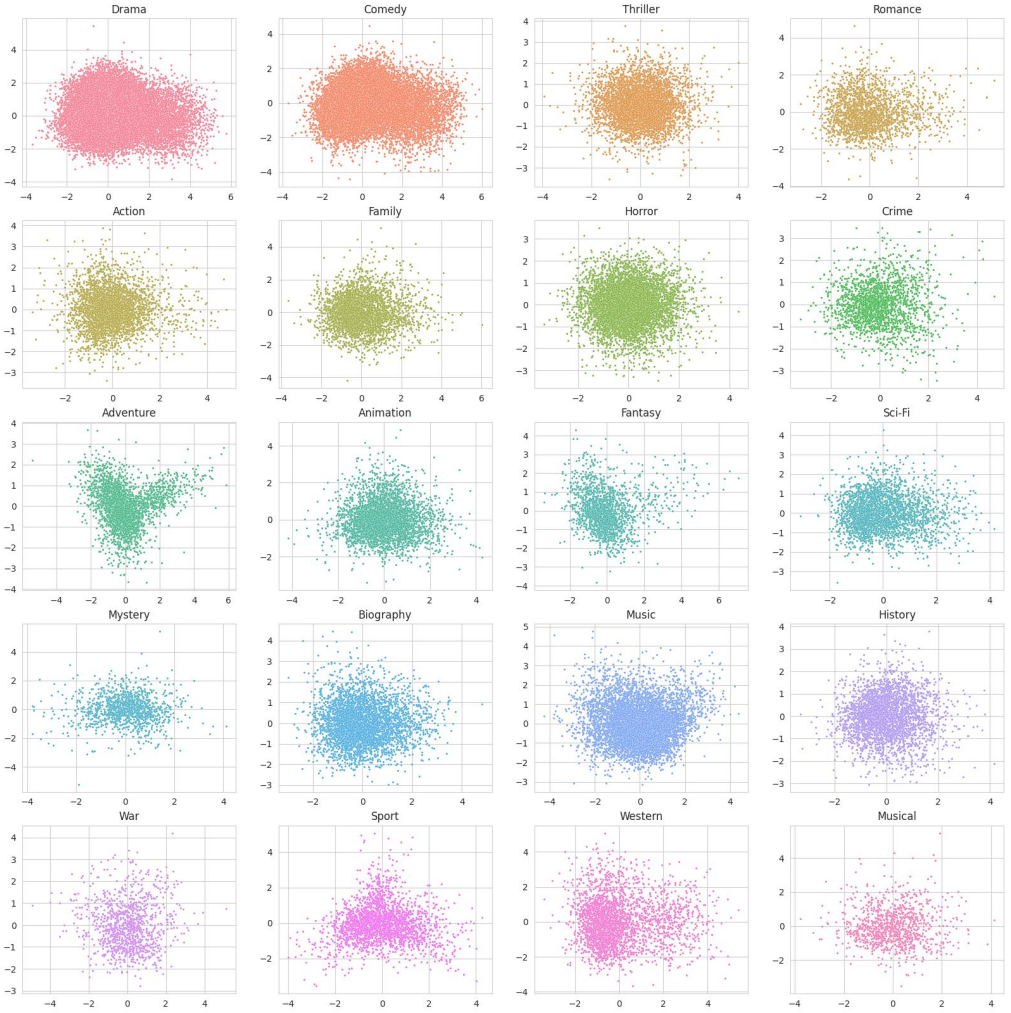
Doc2Vec Model Training



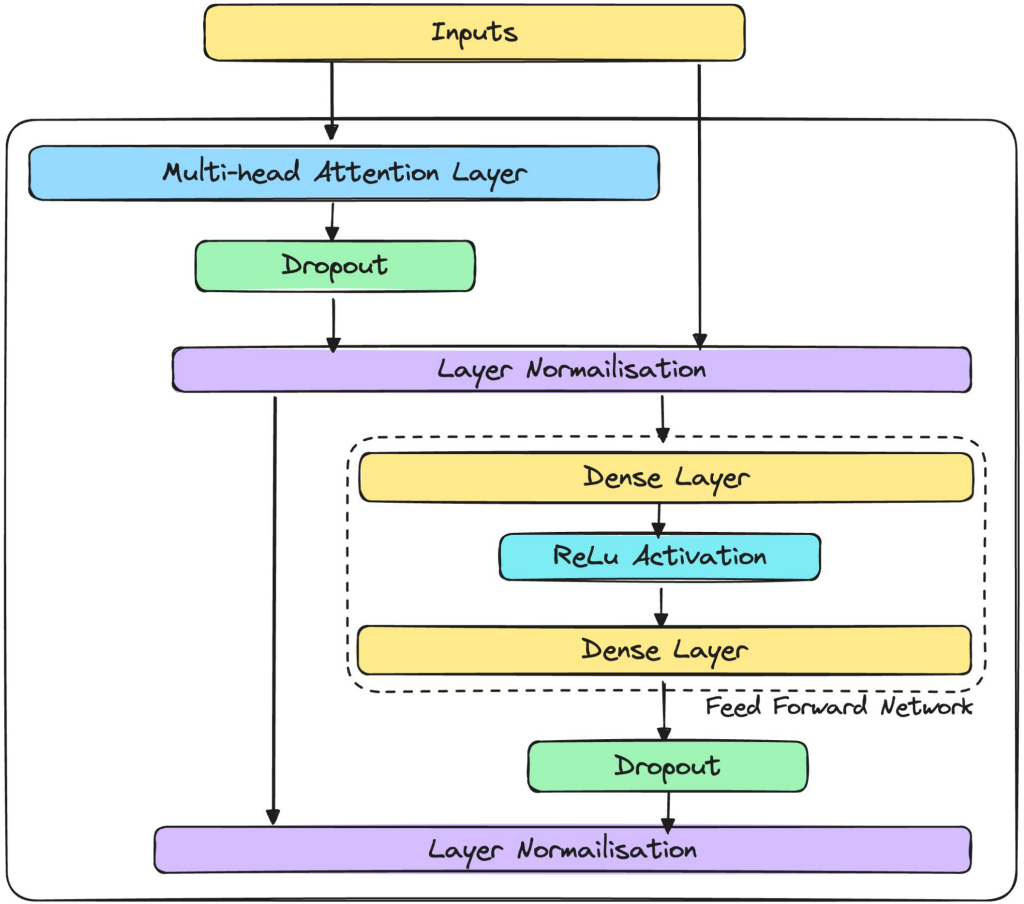
Doc2Vec Embeddings



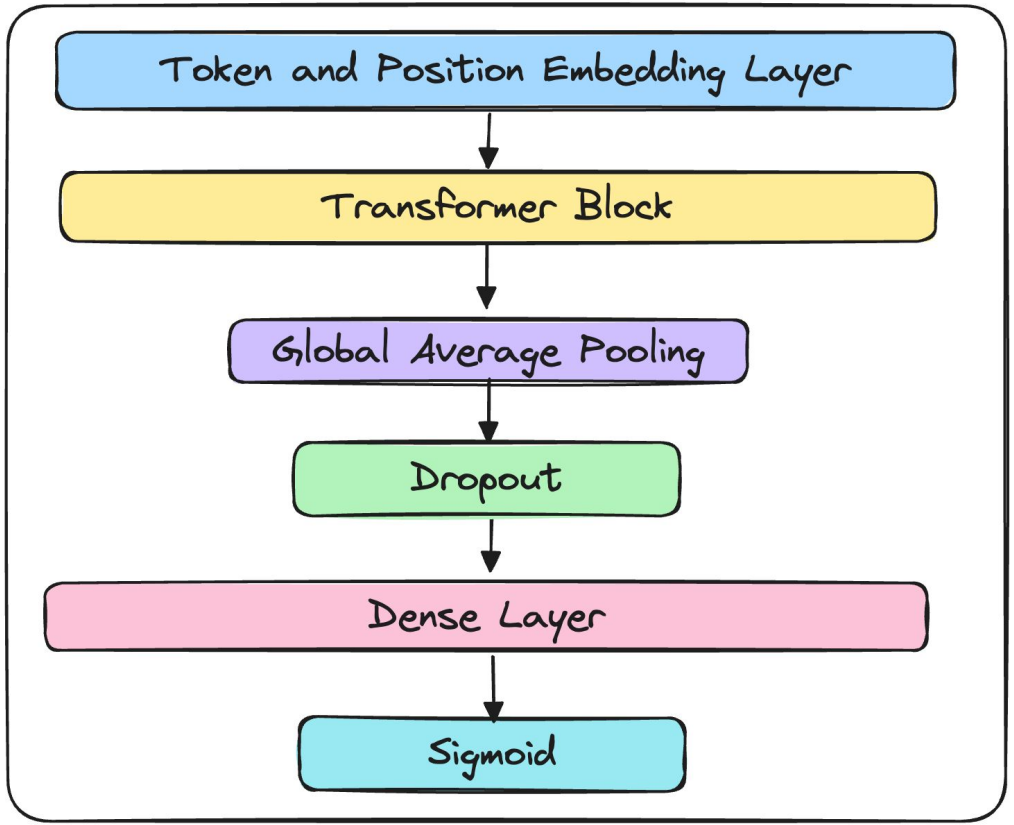
Doc2Vec Embeddings



Transformer Block



Transformer



Results

Metrics used

Jaccard Score: The number of correctly predicted labels divided by the union of predicted and true labels

Hit Rate: Proportion of examples where the model predicts at least one correct genre

F1 Score

Precision

Recall

Accuracy

Hamming Score

Standout Results

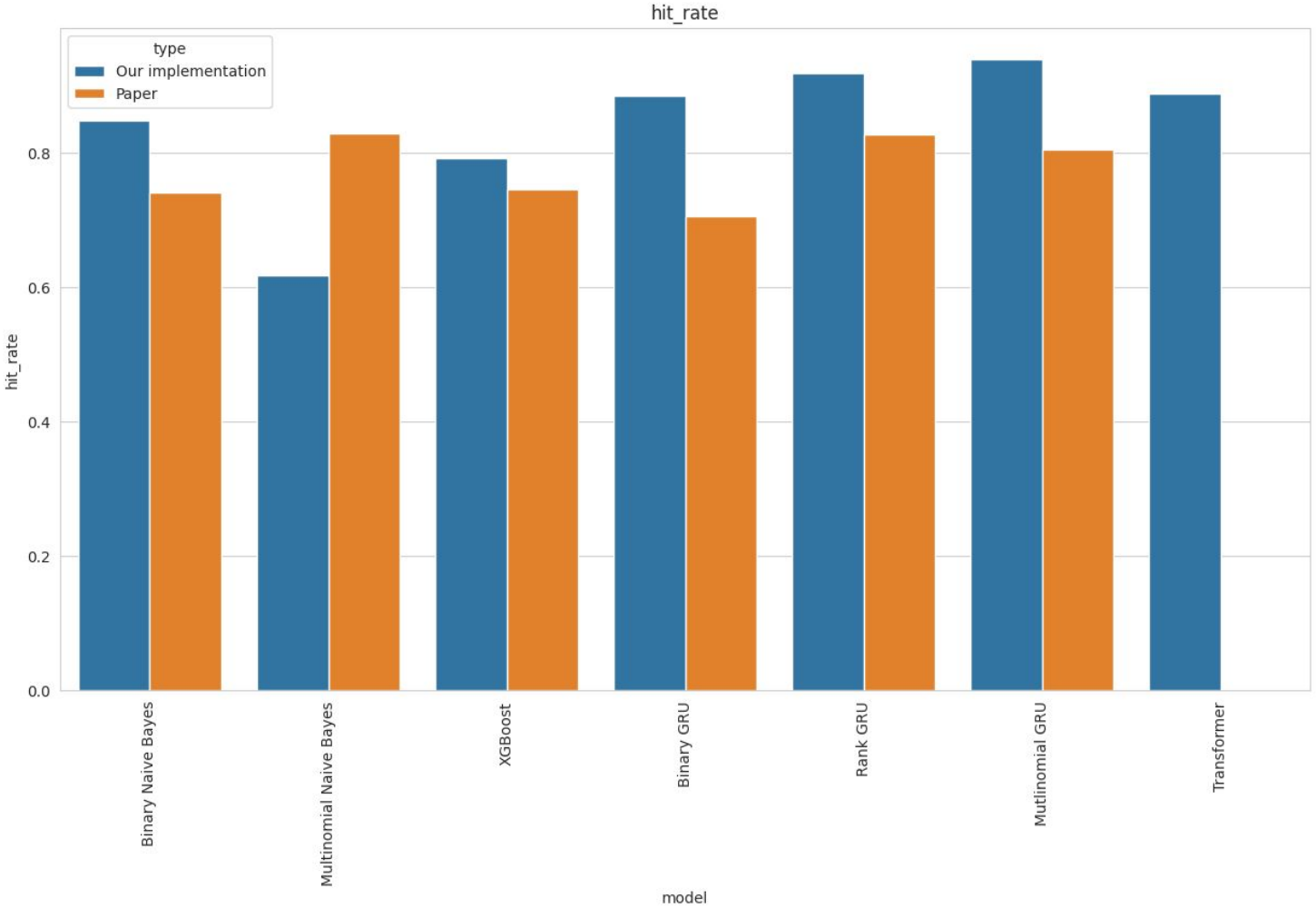
32.5%

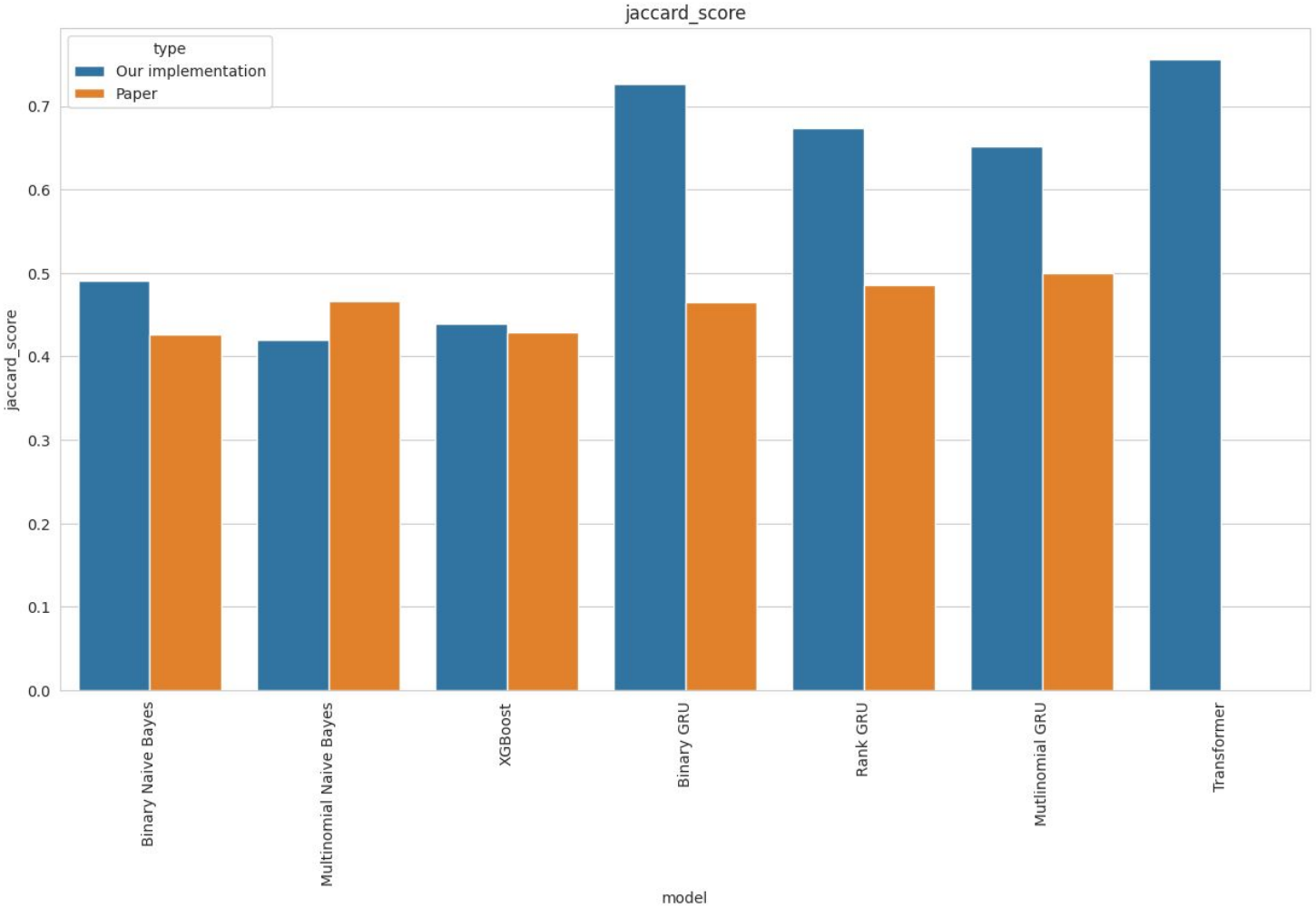
Increase in Jaccard Score for XGBoost with TF-IDF Weighted Word2Vec embeddings over XGBoost with Word2Vec implemented in the paper

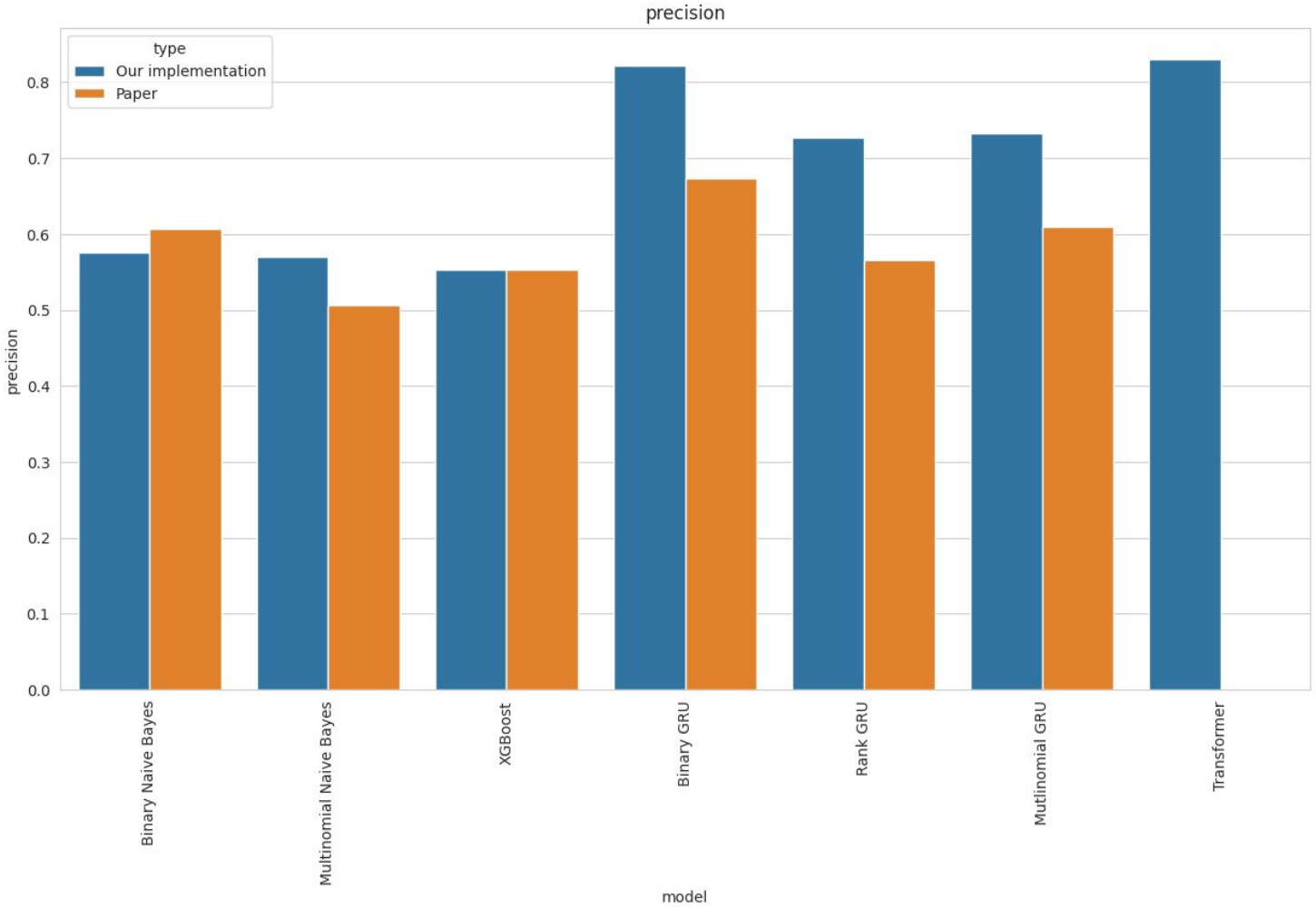
50%

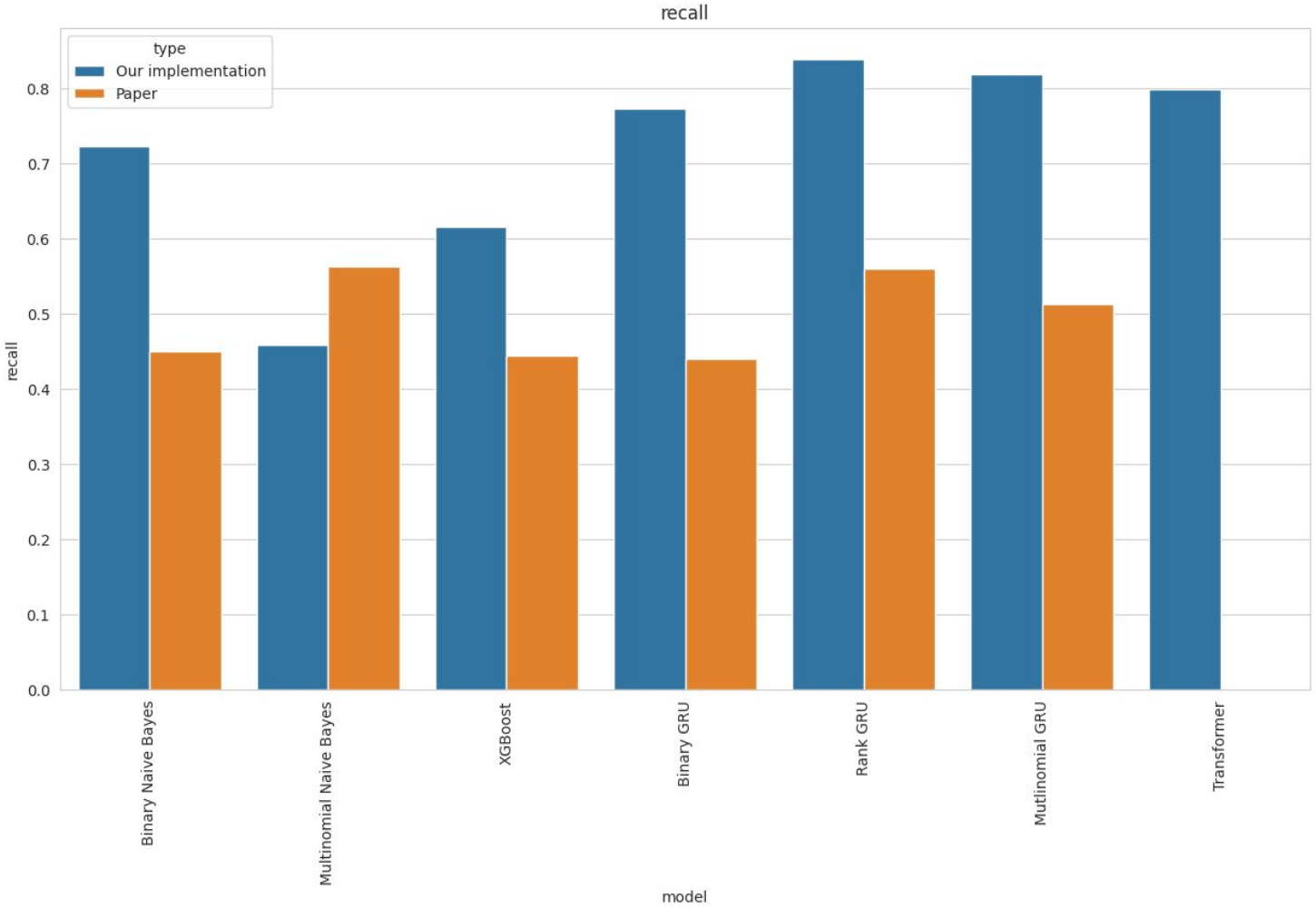
Increase in Jaccard Score for Transformer over Multinomial GRU implemented in the paper

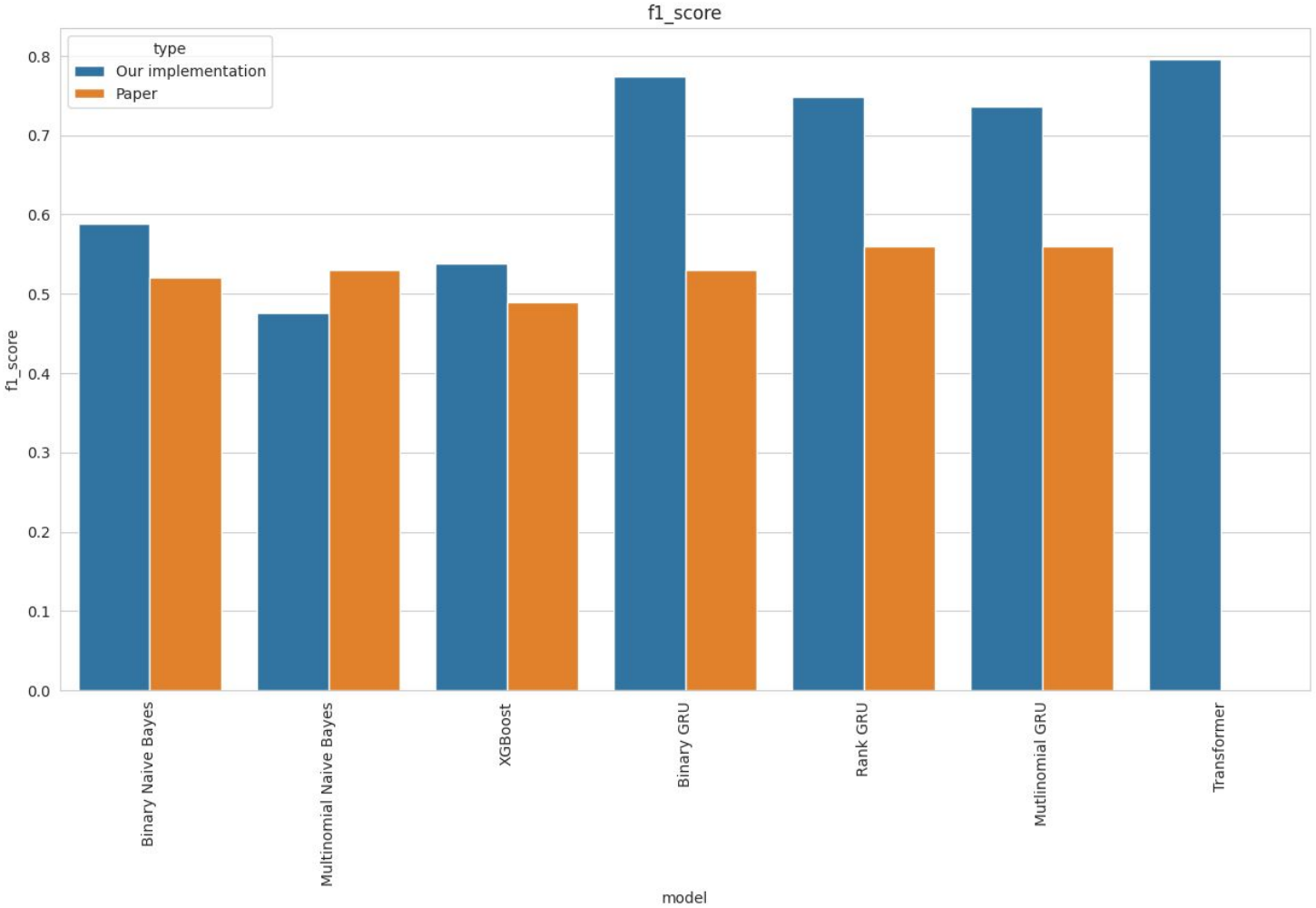
Standard Results

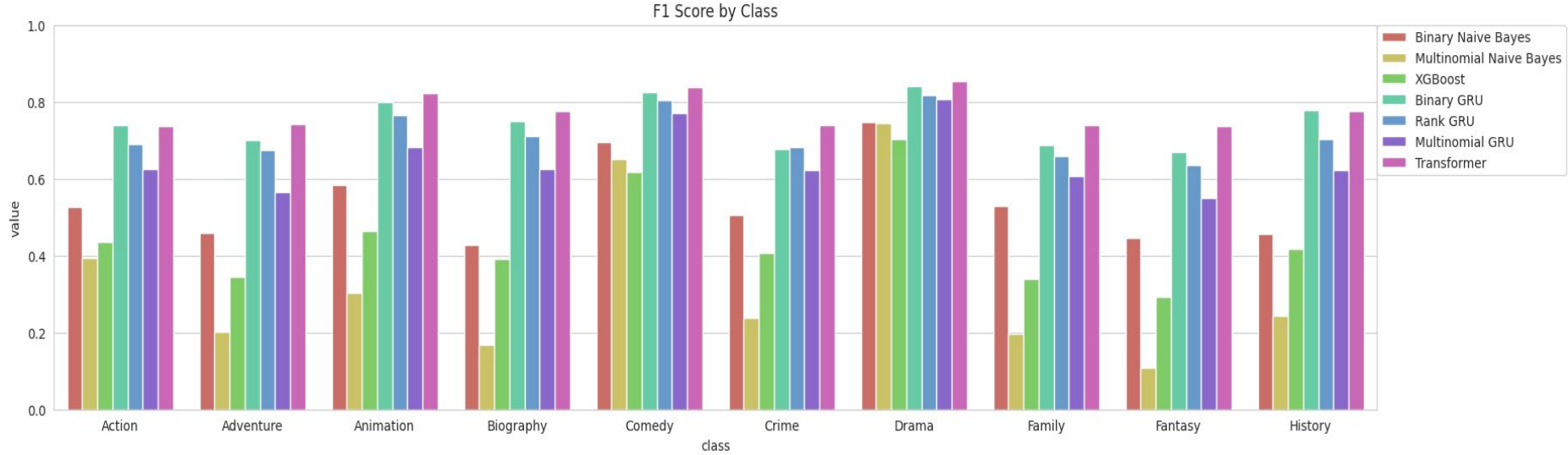

hit_rate
type
Our implementation
Paper
hit_rate
0.0
0.2
0.4
0.6
0.8
Binary Naive Bayes
Multinomial Naive Bayes
XGBoost
Binary GRU
Rank GRU
Mutlinomial GRU
Transformer
model


jaccard_score
type
Our implementation
Paper
jaccard_score
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Binary Naive Bayes
Multinomial Naive Bayes
XGBoost
Binary GRU
Rank GRU
Mutlinomial GRU
Transformer
model

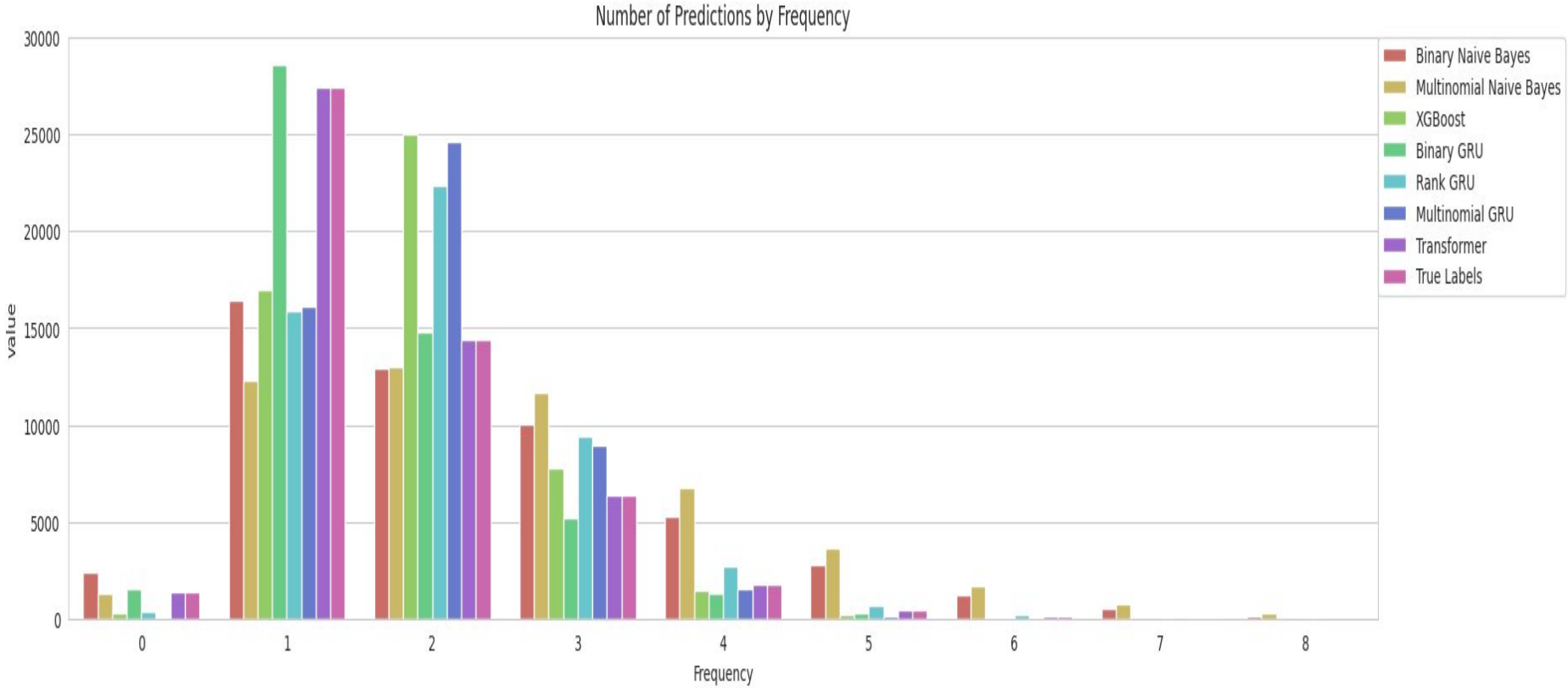

precision
type
Our implementation
Paper
precision
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Binary Naive Bayes
Multinomial Naive Bayes
XGBoost
Binary GRU
Rank GRU
Mutlinomial GRU
Transformer
model


recall
type
Our implementation
Paper
recall
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Binary Naive Bayes
Multinomial Naive Bayes
XGBoost
Binary GRU
Rank GRU
Mutlinomial GRU
Transformer
model


f1_score
type
Our implementation
Paper
f1_score
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Binary Naive Bayes
Multinomial Naive Bayes
XGBoost
Binary GRU
Rank GRU
Mutlinomial GRU
Transformer
model


F1 Score by Class
1.0
0.8
0.6
0.4
0.2
0.0
value
Action
Adventure
Animation
Biography
Comedy
Crime
Drama
Family
Fantasy
History
class
Binary Naive Bayes
Multinomial Naive Bayes
XGBoost
Binary GRU
Rank GRU
Multinomial GRU
Transformer


F1 Score by Class
1.0
0.8
0.6
0.4
0.2
0.0
value
Horror
Music
Musical
Mystery
Romance
Sci-Fi
Sport
Thriller
War
Western
class


Number of Predictions by Frequency
30000
25000
20000
15000
10000
5000
0
value
0
1
2
3
4
5
6
7
8
Frequency
Binary Naive Bayes
Multinomial Naive Bayes
XGBoost
Binary GRU
Rank GRU
Multinomial GRU
Transformer
True Labels

Thank you