

RATING PREDICTION USING CUSTOMER REVIEWS

A Comparative Analysis of Various
Models and their Generalizability

INTRODUCTION

Context, dataset and task

MODEL DEVELOPMENT

Data preparation, model training and
hyperparameter tuning

MODEL GENERALIZATION

Generalizability of the best model

CONCLUSION

Future direction and improvement

SUMMARY

INTRODUCTION

CONTEXT - WHY STUDY



Help to understand customer's like and dislike in order to make better recommendation system

Identify potential issues of the review product (Restaurant, movie and app, etc.)

Measure customer satisfaction

TASK & OBJECTIVE

Task:

- Supervised learning

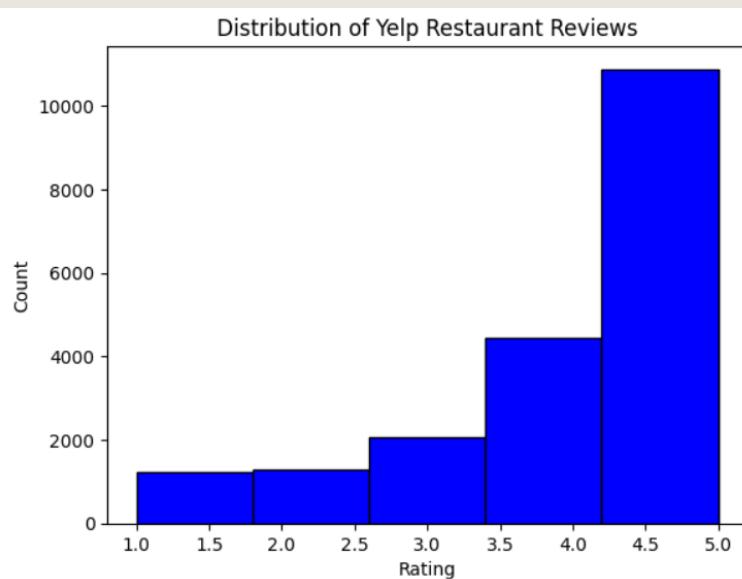
Objective:

- Find the best machine learning model with the lowest MSE on first dataset for prediction rating based on reviews
- Test the best model's generalizability on second dataset

DATASET

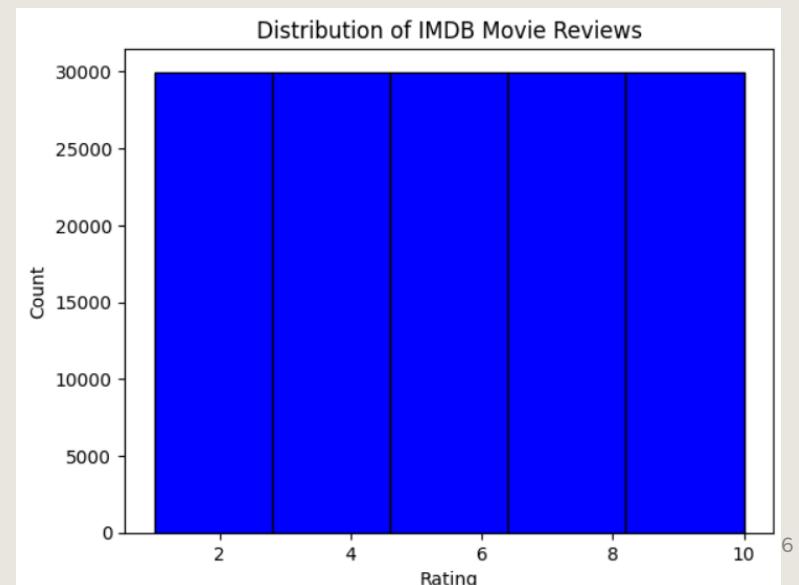
Yelp Restaurant Reviews

- 19 896 Data points
- 1-5 Rating scale, from 1, the lowest rating, to 5, the highest rating
- Dataset for **model training and selection**



IMDB Movie Reviews

- 150 000 Data points
- 1-10 Rating scale, from 1, the lowest rating, to 10, the highest rating
- Dataset for **model generalization**

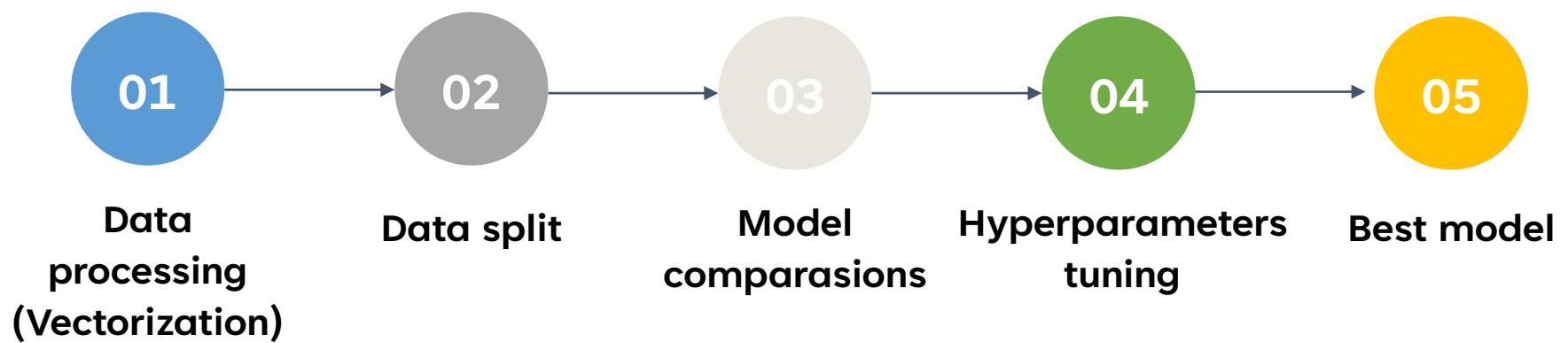


LITERATURE REVIEW

Author(s)	Objectives	Findings	Limitations	Method
Ahmed & Ghabayen., 2020	Improve the accuracy of rating predictions	Prediction framework using deep learning that outperforms existing methods in predicting ratings for online reviews	Depend on the quality and quantity of data used, problem of generalizability	Bidirectional GRU
Sadiq et al., 2021	More accurate prediction of rating to help user's decision making	Use deep learning models to predict the authenticity of numeric ratings and detecting discrepancies between user reviews and numeric ratings	Dataset used is imbalanced in terms of few app categories. Models like LSTM, RNN, GRU, and Bi-LSTM do not perform well on the imbalanced dataset.	CNN, RNN, LSTM, Bidirectional LSTM, GRU

MODEL DEVELOPMENT

PIPELINE



01

DATA PROCESSING

- NLTK:
 - Remove punctuation and stopwords
 - **Tokenize** the text
 - **Lemmatize** the text
 - Join it back into string

- TENSORFLOW:
 - Tokenize the text
 - Convert the text to sequences of integers
 - Pad all sequences in order to ensure a fixed length

Tokenization: separating texts into smaller units, ex: “I like ML” → “I/like/ML”

Lemmatization: grouping different forms of the same word, ex: “drove”, “driving” → “drive”

02

DATA SPLIT

- `Train_test_split` from `sklearn.model_selection`
- **60%** of data points as train dataset
- **40%** of data points as validation dataset

```
# Split the preprocessed data into train and validation sets
y = restaurant["Rating"]
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.4,
random_state=0)
```

03

MODEL COMPARISONS

METHODS USED:

- Average rating (**Baseline**)
- Linear regression
- Bidirectional RNN
- Bidirectional LSTM
- Bidirectional GRU
- CNN

EVALUATION METRIC:

- MSE on validation set

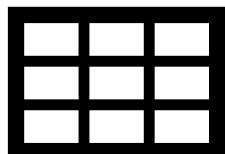
03

MODEL COMPARISONS

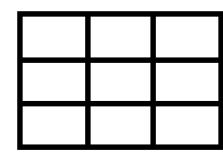
AVERAGE RATING (**BASELINE**)

- Predicting the ratings in the validation set with the average ratings from the train set

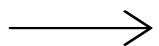
Train set



Validation set



\bar{y}



\hat{y}

03

MODEL COMPARISONS

LINEAR REGRESSION

- Simple linear regression to predict the ratings
- Train the model on the train set
- Prediction on the validation set

```
# Train a linear regression model on the training set
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)

# Make predictions on the validation set
lr_predictions = lr_model.predict(X_val)

# Calculate the MSE of the linear regression model
lr_mse = mean_squared_error(y_val, lr_predictions)

print("Linear regression MSE:", lr_mse)
```

03

MODEL COMPARISONS

BIDIRECTIONAL RNN

- Combination of two RNNs: one move forward and the other move backward
- Single layer of recurrent neurons
- Use output of the previous step and current input to predict next output
- Good to process sequential data (When the points in the dataset are dependent on the other points in the dataset)

03

MODEL COMPARISONS

BIDIRECTIONAL LSTM

- Special kind of RNN capable of learning long-term dependencies
- Has three gates
 - Input gate
 - Output gate
 - Forget gate
- Train the neural by allowing the input to flow in both directions
- Disadvantage: Short-term memory

03

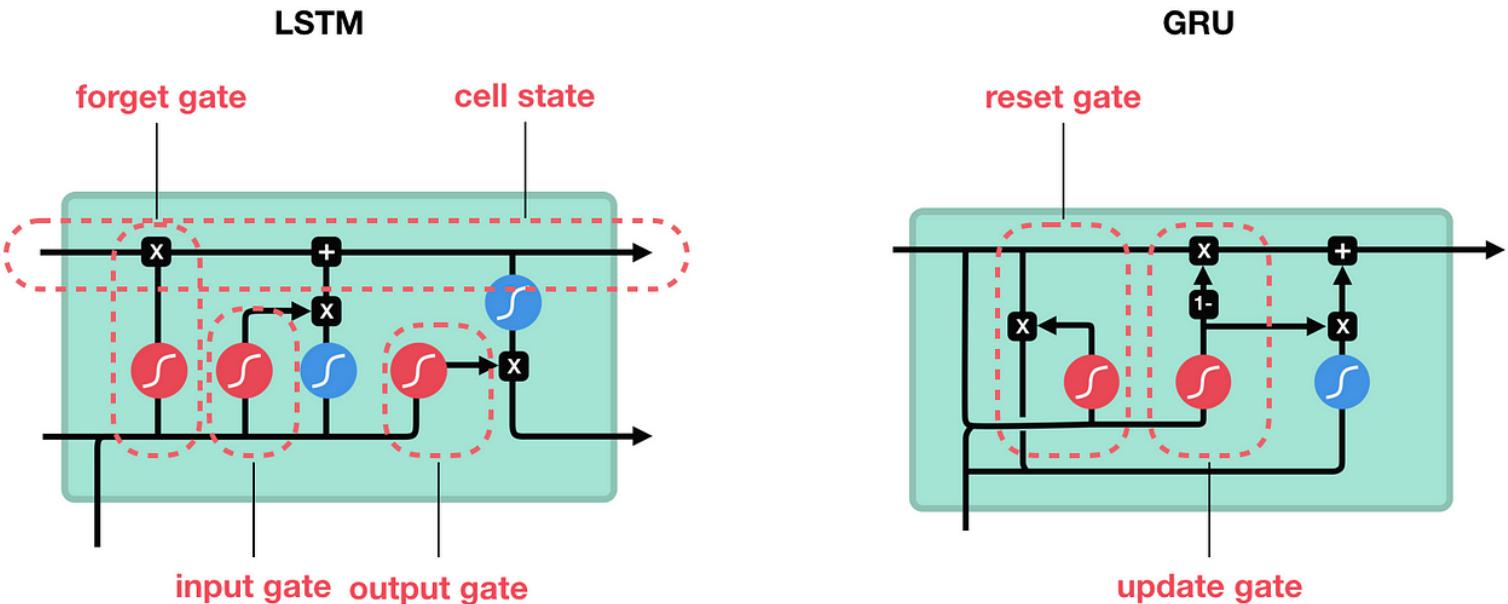
MODEL COMPARISONS

BIDIRECTIONAL GRU

- Variation of the LSTM in term of structure and performance
- Has two gates
 - Update gate
 - Reset gate
- Use of information from previous iteration and next iteration to make predictions about current iteration

03

MODEL COMPARISONS



sigmoid



tanh



pointwise
multiplication



pointwise
addition



vector
concatenation

03

MODEL COMPARISONS

CNN

- Used to perform feature extraction on the data
 - Presence of certain words
 - Order of words
 - Frequency of certain words
- Can handle input sequences of varying lengths

03

PERFORMANCE

	Model	MSE
01	Average (Baseline)	1.4340
02	Linear regression	1.3549
03	Bidirectional RNN	1.0148
04	Bidirectional LSTM	0.6064
05	Bidirectional GRU	0.6561
06	CNN	0.7058

04

HYPERPARAMETERS TUNING

HYPERPARAMETERS TUNED:

- **Embedding dim:** min_value=60, max_value=120, step=20
- **Lstm units:** min_value=32, max_value=96, step=32
- **Dropout rate:** min_value=0.1, max_value=0.5, step=0.1
- **L2 rate:** min_value=0.01, max_value=0.05, step=0.01

PACKAGE USED:

- **RandomSearch** from **KerasTuner**
- Core Code:

```
# Define the hyperparameter search space
tuner = RandomSearch(lstm_model, objective='val_loss',
                      max_trials=10, executions_per_trial=3,
                      directory='hyperparameter_tuning',
                      project_name='yelp_reviews_lstm')
```

04

PERFORMANCE AFTER TUNING

Model	Parameter	Best parameter	MSE
Baseline	NA	NA	1.4436
LSTM	Embedding dim	80	
LSTM	Lstm units	96	
LSTM	Dropout rate	0.3	0.5851
LSTM	L2 rate	0.01	

Model	MSE before tuning
LSTM	0.6064

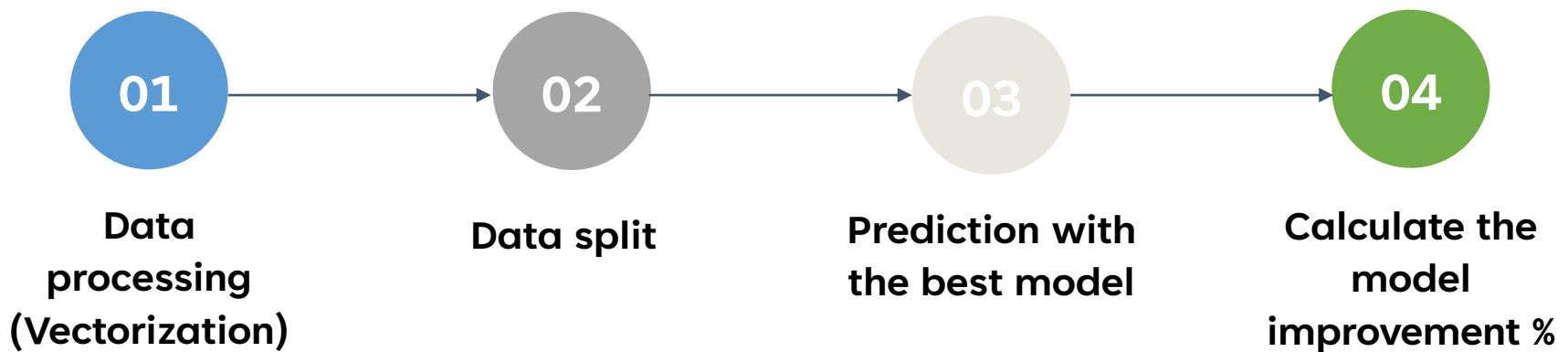
05

BEST MODEL

- The **LSTM model** shows a superior performance compared to other models in our project.
- **LSTM model** captures well the context and meaning of words in sequential data
- **CNN** and **GRU** models didn't perform as well as the papers suggest in our project, but they still produced relatively good results than the baseline.

MODEL GENERALIZATION

PIPELINE



RESULT ANALYSIS

CALCULATION FORMULA

$$(\text{Baseline_MSE} - \text{LSTM_MSE}) / \text{Baseline_MSE} * 100\%$$

RESULT COMPARISON

	Baseline	LSTM	Improvement
Restaurant review	1.4334	0.5851	59.18%
Movie review	8.2951	4.0056	51.71%

CONCLUSION

FUTURE DIRECTION & IMPROVEMENT

- Continue hyperparameter tuning
- Explore different models
- Incorporate sentiment analysis into the rating prediction model by initially dividing the dataset into two separate sets based on positive and negative sentiment. Then, the rating prediction can be performed for both datasets (Qiu et al., 2018 & Ahmed & Ghabayen., 2020)

Author(s)	Objectives	Findings
Qiu et al., 2018	Predictive framework based on sentiment analysis for ratings prediction	Implement sentiment analysis at the aspects-level can improve prediction accuracy

REFERENCE

- Ahmed, B. H., & Ghabayen, A. S. (2020). Review rating prediction framework using Deep Learning. *Journal of Ambient Intelligence and Humanized Computing*, 13(7), 3423–3432.
<https://doi.org/10.1007/s12652-020-01807-4>
- Qiu, J., Liu, C., Li, Y., & Lin, Z. (2018). Leveraging sentiment analysis at the aspects level to predict ratings of reviews. *Information Sciences*, 451-452, 295–309.
<https://doi.org/10.1016/j.ins.2018.04.009>
- Sadiq, S., Umer, M., Ullah, S., Mirjalili, S., Rupapara, V., & Nappi, M. (2021). Discrepancy detection between actual user reviews and numeric ratings of Google App Store using Deep learning. *Expert Systems with Applications*, 181, 115111.
<https://doi.org/10.1016/j.eswa.2021.115111>
- *Understanding LSTM networks.* Understanding LSTM Networks -- colah's blog. (n.d.). Retrieved April 11, 2023, from <https://colah.github.io/posts/2015-08-Understanding-LSTMs/?fbclid=IwAR3qLdQD3b1mP4jh4SNQEAE58upxTstEqXyjg5w-5TbXK8zWTMu1z5seXc>

THANK YOU

Xiao LI

Landry KUBWIMANA



ÉCOLE DE
TECHNOLOGIE
SUPÉRIEURE
Université du Québec

FAKE NEWS

Automatic Fake News Detection

Event: Machine Learning – MATH80629A | MSc. Student: Faramarz Farhangian | April 14th, 2023 (Quebec, CA)

Presented as Final Project, Department of Decision Science, HEC

■ Agenda

1- Introduction

- 1.1- Motivations
- 1.2- Fake news and Related Concepts
- 1.3- Fake news features
- 1.4- Fake news perspectives
- 1.5- Research objectives

2- Comparative study

- 2.1- Research objectives
- 2.2- Datasets
- 2.3- Text Pre-processing
- 2.4- Text Representation
- 2.5- Classification Algorithms
- 2.6- Evaluation Metrics
- 2.7- Experimental setting

3- Results

- 3.1- Summary Results
- 3.2- Complexity

4- Conclusion and Discussion

5- References

6- Question time

Part 1:

Introduction

■ 1. Introduction

1.1. Motivations

- ❖ While **fake news is not a new concept**, today's getting more important with the **widespread use of social media**.
- ❖ Social media characteristics: **Low cost, Easy access, Rapid circulation**
- ❖ These characteristics pave the way for **profiteers** to spread fake news with **political and economic motives**.
- ❖ Fake news are problematic and make a real-world consequences like U.S. Presidential election or Nuclear war!
- ❖ human ability to distinguish fake news from true news is **poor**



■ 1. Introduction

1.2. Fake news and Related Concepts

Concept	Authenticity	Intention	News?
Deceptive news	Non-factual	Mislead	Yes
False news	Non-factual	Undefined	Yes
Satire news	Non-unified ²	Entertain	Yes
Disinformation	Non-factual	Mislead	Undefined
Misinformation	Non-factual	Undefined	Undefined
Cherry-picking	Commonly factual	Mislead	Undefined
Clickbait	Undefined	Mislead	Undefined
Rumor	Undefined	Undefined	Undefined

The following concepts are different from fake news based on our definition:

- **Satire news** that has no intent to mislead
- **Rumors** that did not originate from news events
- **misinformation** that is created unintentionally
- **hoaxes** that are only motivated by fun

What Fake news is?

Definition 1.1 (Broad definition of fake news). Fake news is false news.

Definition 1.2 (Narrow definition of fake news) Fake news is a news article that is intentionally and verifiably false.

We consider the second definition as the main definition because:

1. It is more **comprehensive**
2. **can distinguish fake news from other concepts.**

■ 1. Introduction

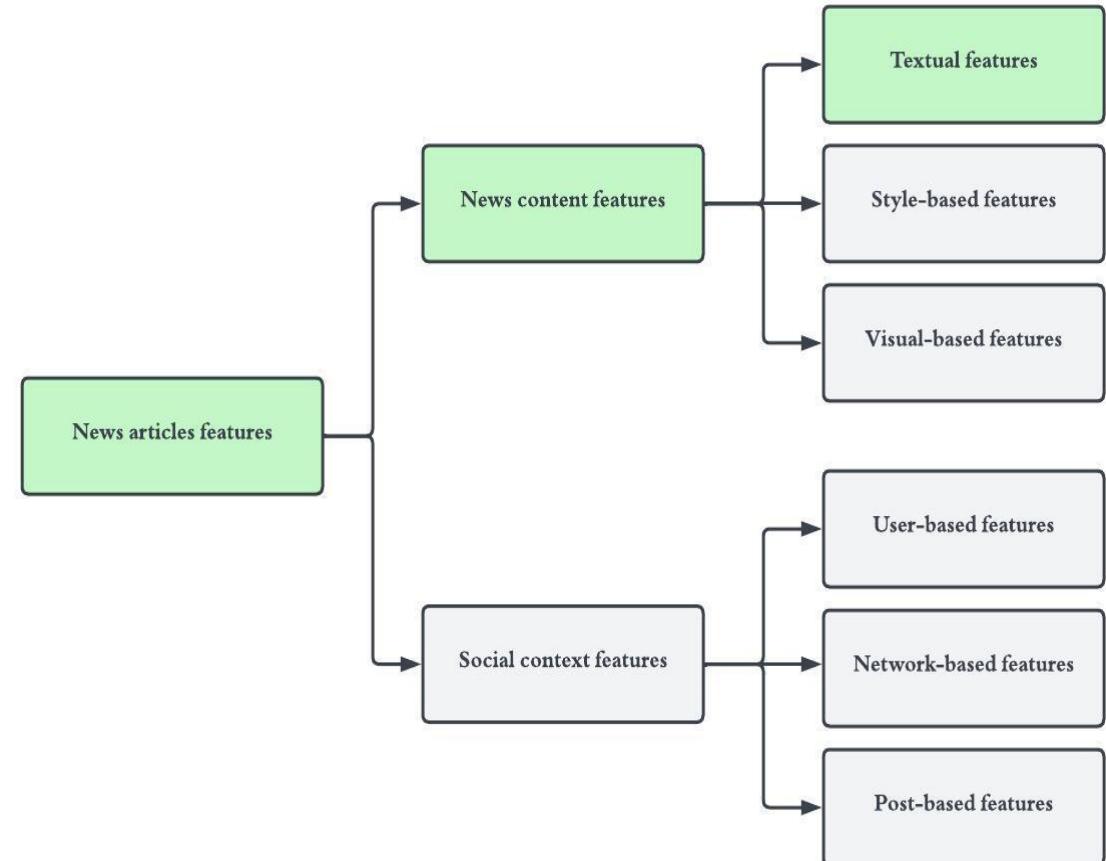
1.3. Fake news features

How news content features help?

- News content features describe the meta information related to a piece of news. For example **headlines, body text, image or videos.**

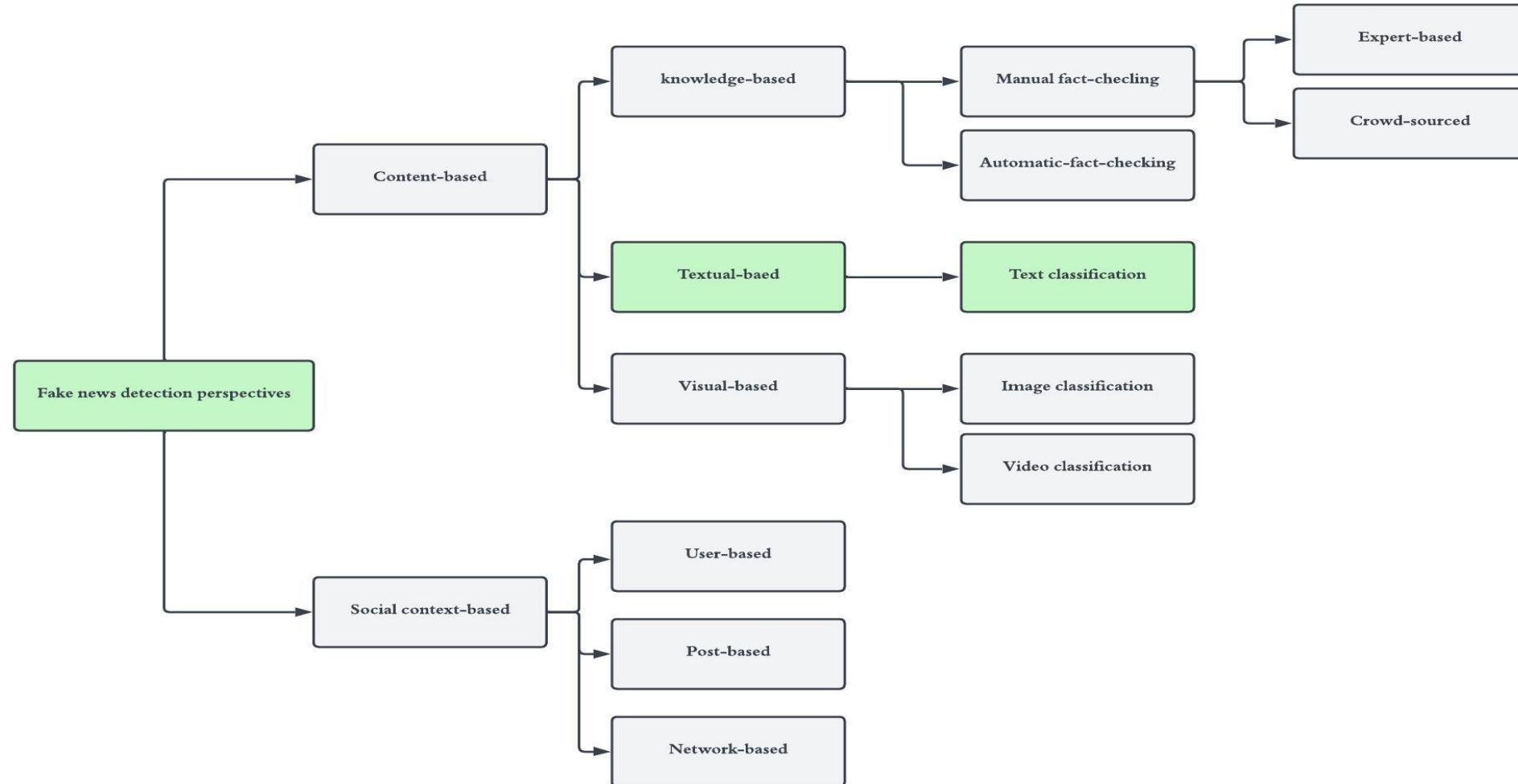
How social context features help?

- Social context refers to the entire social environment in which the dissemination of the news operates, **including how the social data is distributed** and **how online users interact with each other.**
- **users, generated posts, other users express their opinions and emotions** through posts/comments, and these users came from different **types of networks** on social media.



1. Introduction

1.4. Fake news perspectives



Part 2:

Comparative Study

■ 2. Comparative Study

2.1. Research objectives

Definition 1.3 (Fake News Detection) Given the a news article a , the task of fake news detection is to predict whether the news article a is fake or not.

$$f(a) = \begin{cases} 0, & \text{If } a \text{ is fake} \\ 1, & \text{Otherwise} \end{cases}$$

- ❖ In this study we presents an empirical comparison of state-of-the-art text classifiers and representation techniques using two benchmark datasets under the same experimental protocol.
- ❖ The study considers fake news detection as a text classification task and follows five phases: Preparing labeling datasets, Pre-processing dataset, Text representation phase, Training classification models, Evaluation.

Research questions

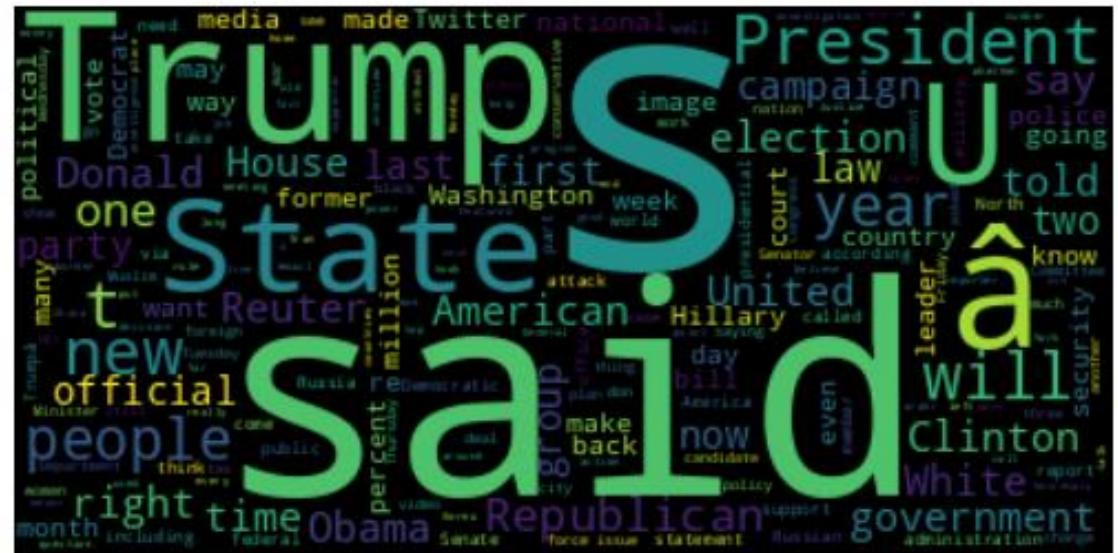
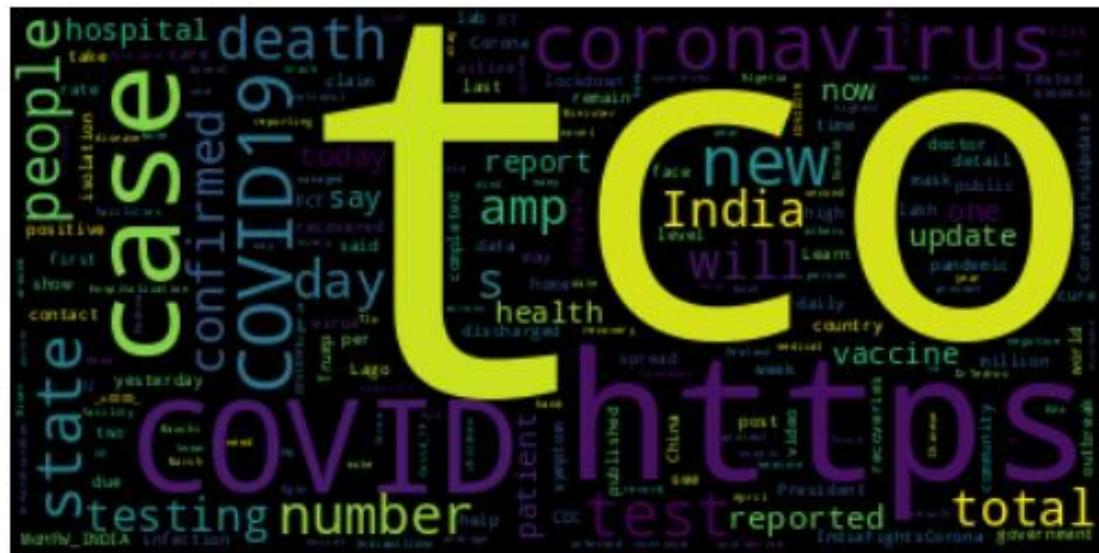
- ***What machine learning models are commonly used for this task?***
- ***What are the various feature extraction methods employed in this task?***
- ***Which models perform best in the same experimental setting?***

2. Comparative study

2.2. Datasets

This study uses two benchmark datasets from *various domains to prevent research bias*

Dataset	Domain	Media	Fact-checking	Size	No. Class	Distribution
ISOT	Politics & World news	Mainstream media	Fact-checking websites (expert)	44866	2	(23448,21417)
COVID	Covid-19 & Health	Twitter	Fact-checking websites (expert)	10700	2	(5100,5600)



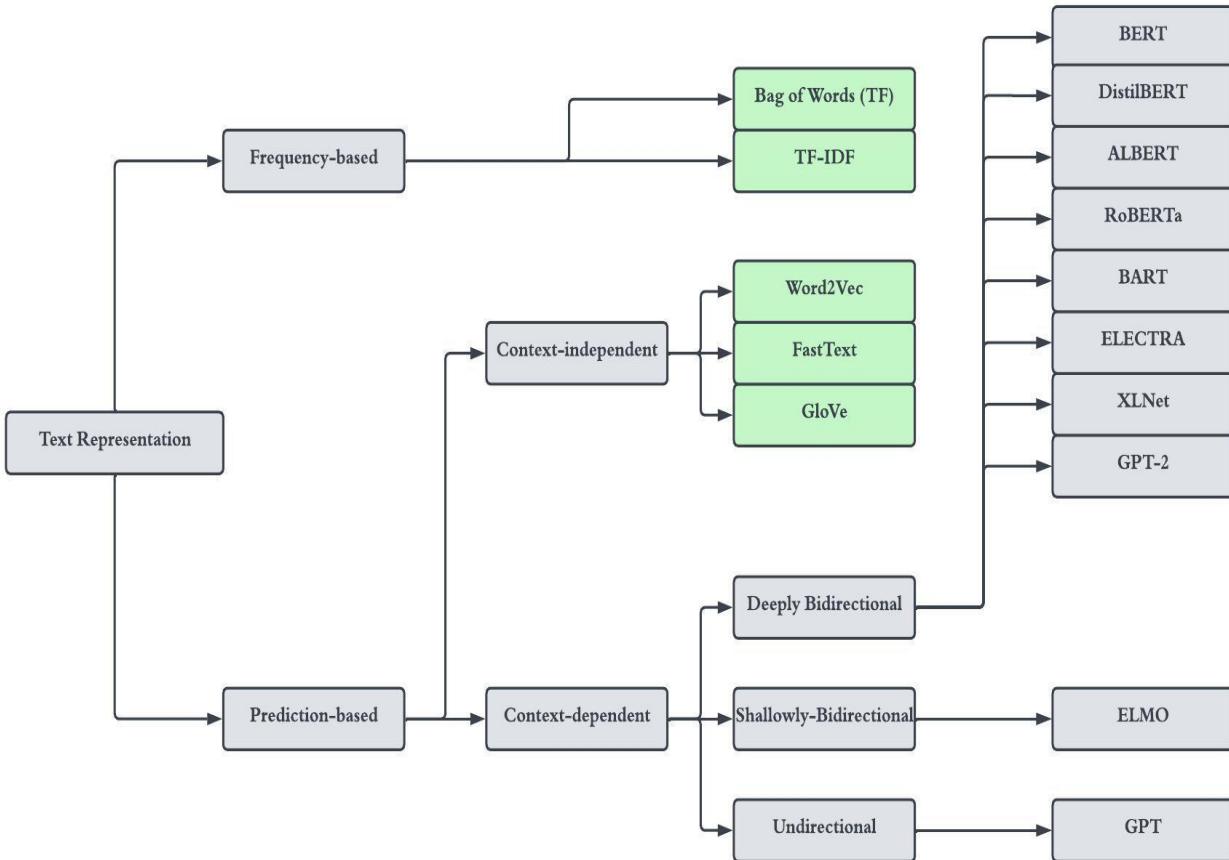
■ 2. Comparative study

2.3. Text Pre-processing



■ 2. Comparative study

2.4. Text Representation



Frequency-based

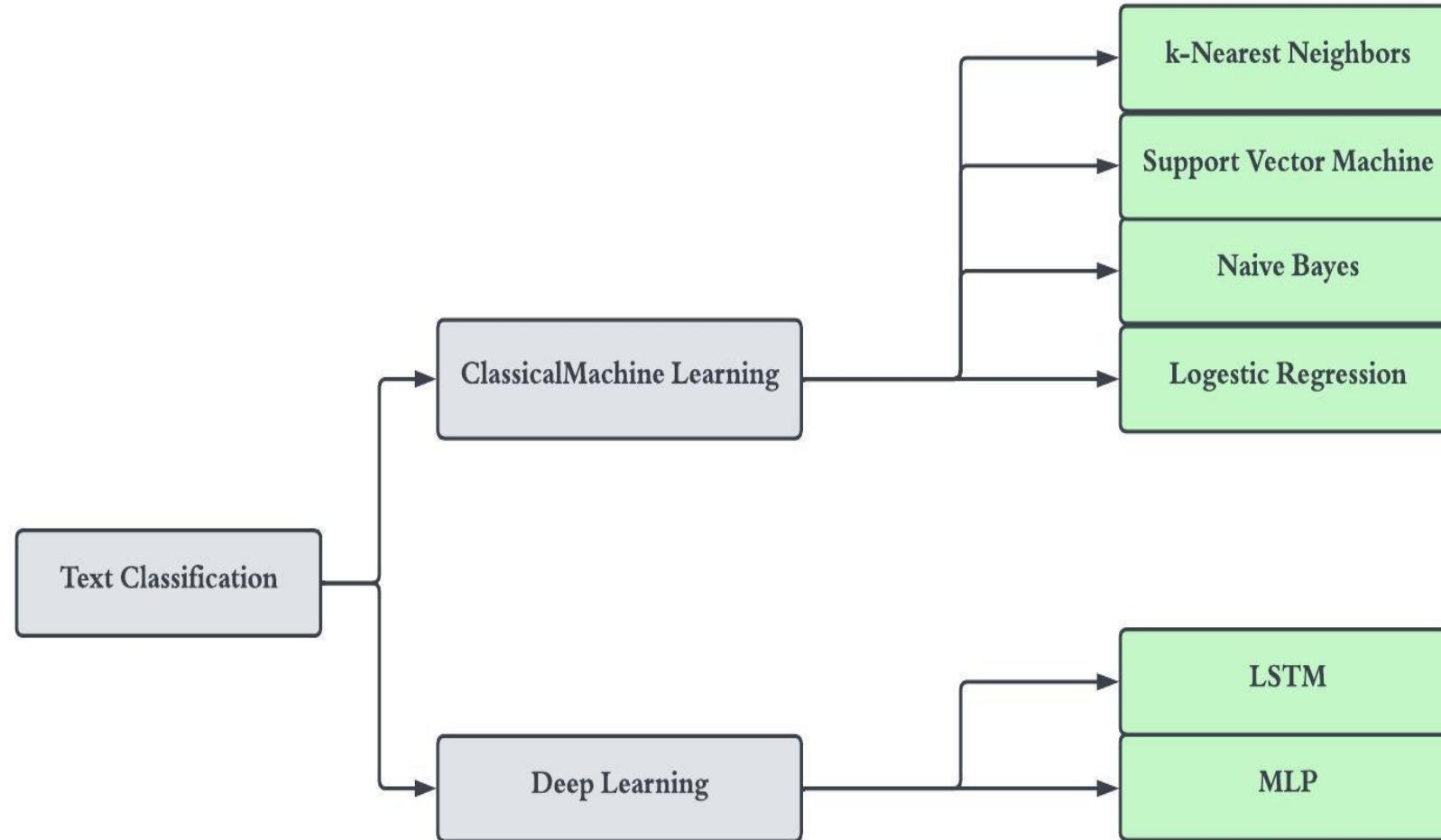
- Focusing on occurrence or co-occurrence of words in a document.
- Very simple and popular
- They do not represent useful information about the meaning and relations between words.
- Bag of Words Or Term Frequency just gives the information about the frequency of words in document and say noting about positions and structure of words also say noting about word importance!
- TF-IDF: scoring and weighting words in order to find the relevant and important words in a document.

Prediction-based

- Create numerical vectors from textual data which has known as word embedding
- Can extract semantic and syntactic features of words.

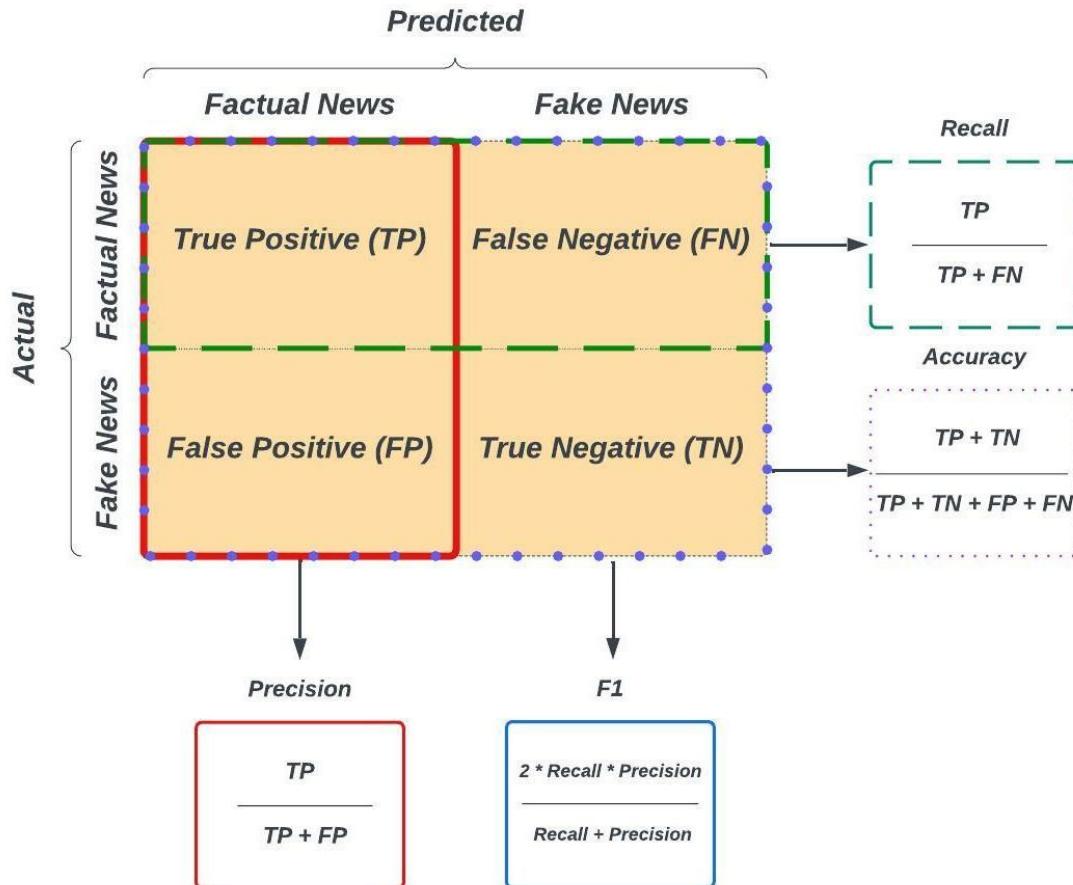
■ 2. Comparative study

2.5. Classification Algorithms



■ 2. Comparative study

2.6. Evaluation Metrics



- **True Positive (TP)** when a news article is predicted as fake news when it is labeled as fake news.
- **True Negative (TN)** when a news article is predicted as real news when it labeled as real news.
- **False Positive (FP)** when a news article predicted as fake news when it labeled as real news.
- **False Negative (FN)** when a news article predicted as real news when it labeled as fake new.

■ 2. Comparative study

2.7. Experimental setting

Text Representation

- In all word embedding methods, we considered the **average of word feature vectors that occurred in each document** in order to represent the feature vector of each document in datasets.
- **Word2Vec** trained over a google news corpus with 300 features retrieved from each sample.
- **GloVe**, trained over a wikigigaword corpus with 300-dimensional feature vectors.
- **Fasttext** trained over the Wikipedia corpus with 300 features per sample.

Classification

- we conducted hyper-parameter tuning using **grid search** techniques to find the optimal hyper-parameter values.
- We used **5-fold cross-validation**.
- We consider **baseline** based on **most frequent strategy**.
- The experiments were conducted on **Colab Pro**.

Method	Hyperparameters
SVM	Kernel: ['rbf', 'linear', 'poly', 'sigmoid'] Gamma: [1, 0.1, 0.01, 0.001, 0.0001] C: [0.1, 1, 10, 100, 1000]
LR	solver: ['liblinear'] penalty: ['none', 'l1', 'l2', 'elasticnet'] C: [0.01, 0.1, 1, 10, 100]
KNN	n_neighbors: [1 - 20]
MLP	hidden layer sizes: [(32,), (64,), (128,)] activation: ['relu', 'tanh', 'logistic'] solver: ['adam', 'sgd'] batch size: [16, 32, 64] learning rate: [0.001]
LSTM	Activation function: [sigmoid, ReLU] Batch size: [64, 128, 512] Number of epochs: [5,20,100] Optimizer: [Adam] Learning rate: 0.001 Hidden size: 128 Dropout: 0.2

Part 3:

Results

3. Results

3. Results

In this section, we want to answer the following questions:

(RQ1) What machine learning models are commonly used for this task?

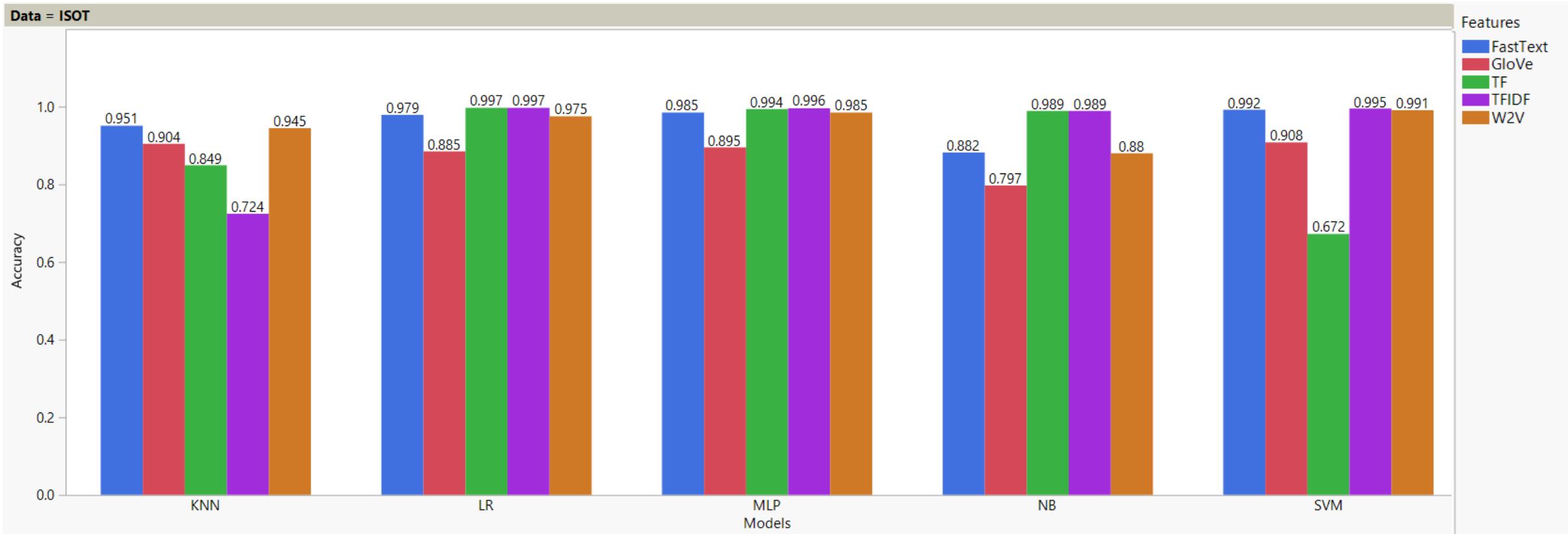
(RQ2) What are the various feature extraction methods employed in this task?

(RQ3) Which models perform best in the same experimental setting?

- ❖ We already answer to the **RQ1**, and **RQ2** in **Text representation**, and **Classification algorithms** sections.
- ❖ Models has the same level of performance in terms of accuracy, precision, recall, and F1-score, which indicates that it is well balanced in predicting both positive and negative classes. In other words, the model's accuracy is high and it can predict both positive and negative classes with equal efficiency.
- ❖ Our baseline performance is 50% because we have balanced data in both dataset.

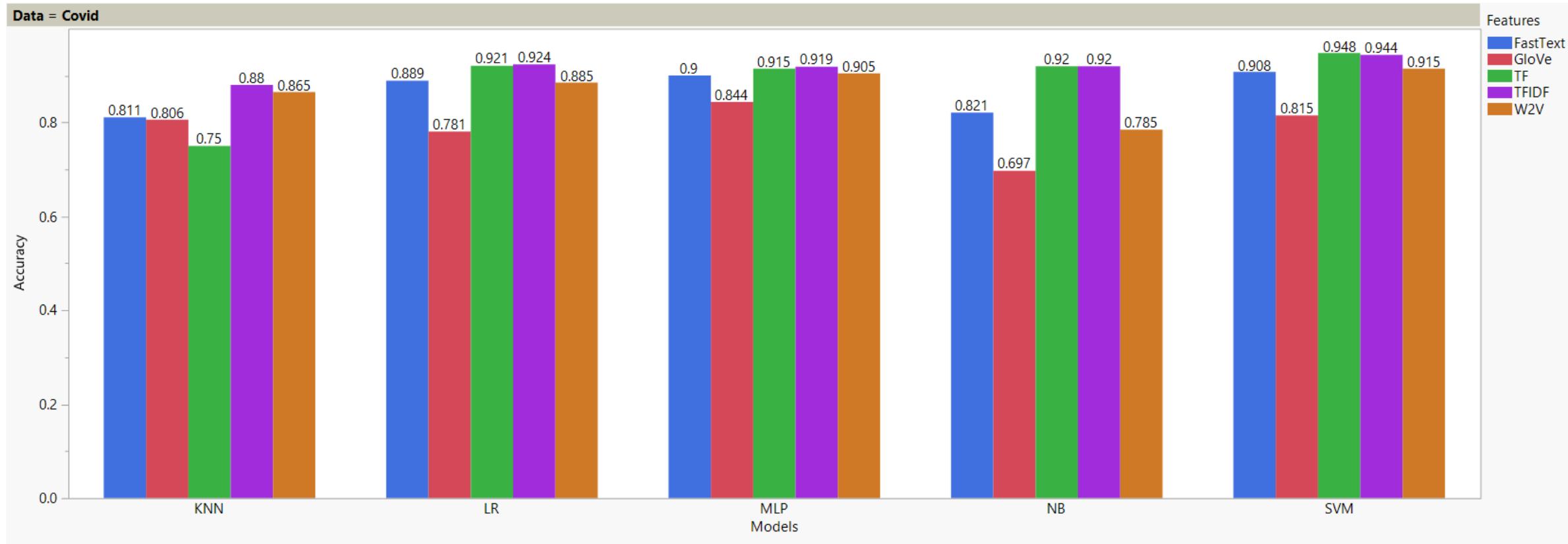
3. Results

3.1. Summary results



3. Results

3.1. Summary results



3. Results

3.2. Complexity

ISOT Dataset										
Models	LR		KNN		NB		SVM		MLP	
Features	Training time	Prediction time								
TF	129.73	0.011	696.17	30.22	1.167	0.074	25061.06	537.18	9026.57	0.192
TFIDF	17.67	0.004	25985.43	3540.55	3.385	0.239	6234.75	185.33	13562.86	0.156
W2V	150.37	0.018	47.2	5.896	2.033	0.098	1736.89	7.065	793.72	0.069
GloVe	170.58	0.003	9.16	1.83	0.464	0.031	899.073	6.65	118.323	0.0225
FastText	1629.17	0.022	37.386	4.169	3.262	0.121	2658.86	5.545	1112.159	0.093

COVID Dataset										
Models	LR		KNN		NB		SVM		MLP	
Features	Training time	Prediction time								
TF	30.79	0.001	6.613	0.774	0.135	0.0046	149.443	1.604	185.243	0.005
TFIDF	0.5	0.0007	27.409	4.178	0.142	0.003	137.517	2.023	206.9	0.004
W2V	109.84	0.029	3.026	1.353	0.55	0.049	649.34	2.232	181.75	0.031
GloVe	4.743	0.001	0.796	0.326	0.125	0.003	72.688	0.6157	68.44	0.004
FastText	125.865	0.002	3.149	0.95	0.8346	0.115	220.23	1.782	390.145	0.0203

Part 4:

Conclusion and Discussion

■ 4. Conclusion and Discussion

- All of the models outperform the baseline in each metrics.
- Frequency-based methods are the best text representation method in this task especially by using classical model.
- In ISOT dataset, SVM with frequency-based methods outperformed the others.
- In COVID dataset LR with frequency-based methods outperformed the others.
- From the word embedding methods, FastText outperformed the others.
- Despite the frequency-based methods outperformed but because of the sparse matrix they have more complexity.
- In classical machine learning models, KNN, SVM are the most complexity, and LR, and NB is the lowest one.
- MLP have more training time but low prediction cost, and FastText, and Word2Vec are more complex than GloVe.
- One of the reason that frequency-based methods outperform the word embedding methods is existence of out of vocabulary problem in word embedding methods.

■ 4. Conclusion and Discussion

- **Multi-modal fake news detection**
- **Fake news detection using Ensemble methods**
- **Fake news detection using weekly supervised methods**
- **Fake news explanation**
- **Fake news detection for different languages like French or Persian**

Part 5:

References

5. References

- 1 K. Shu, A. Sliva, S. Wang, J. Tang, H. Liu, Fake news detection on social media: A data mining perspective, ACM SIGKDD explorations newsletter 19 (1) (2017) 22–36
- 2 N. Diakopoulos, M. De Choudhury, M. Naaman, Finding and assessing social media information sources in the context of journalism, in: Proceedings of the SIGCHI conference on human factors in computing systems, 2012, pp. 2451–2460
- 3 Hermida, Twittering the news: The emergence of ambient journalism, Journalism practice 4 (3) (2010) 297–308
- 4 P. Tolmie, R. Procter, D. W. Randall, M. Rouncefield, C. Burger, G. Wong Sak Hoi, A. Zubiaga, M. Liakata, Supporting the use of user generated content in journalistic practice, in: Proceedings of the 2017 chi conference on human factors in computing systems, 2017, pp. 3632–3644.
- 5 Zubiaga, A. Aker, K. Bontcheva, M. Liakata, R. Procter, Detection and resolution of rumours in social media: A survey, ACM Computing Surveys (CSUR) 51 (2) (2018) 1–36
- 6 A. Bondielli, F. Marcelloni, A survey on fake news and rumour detection techniques, Information Sciences 497 (2019) 38–55.
- 7 J. C. Reis, A. Correia, F. Murai, A. Veloso, F. Benevenuto, Supervised learning for fake news detection, IEEE Intelligent Systems 34 (2) (2019) 76–81
- 8 X. Zhou, R. Zafarani, A survey of fake news: Fundamental theories, detection methods, and opportunities, ACM Computing Surveys (CSUR) 53 (5) (2020) 1–40.
- 9 H. Webb, P. Burnap, R. Procter, O. Rana, B. C. Stahl, M. Williams, W. Housley, A. Edwards, M. Jirotka, Digital wildfires: Propagation, verification, regulation, and responsible innovation, ACM Transactions on Information Systems (TOIS) 34 (3) (2016) 1–23
- 10 H. Karimi, P. Roy, S. Saba-Sadiya, J. Tang, Multi-source multi-class fake news detection, in: Proceedings of the 27th international conference on computational linguistics, 2018, pp. 1546–1557

“Thank You”
for your attention



Question Time!

MATH60629A

MACHINE LEARNING I: LARGE SCALE DATA ANALYSIS AND DECISION MAKING

FEDERATED LEARNING

SIDDHANT ARORA

PHILIPPE BELIVEAU

KARAN JOSEPH

PROBLEM STATEMENT

The central research question is to compare the performance of centralized Machine Learning model, Federated Learning model, and Federated Learning with Differential Privacy (DP)

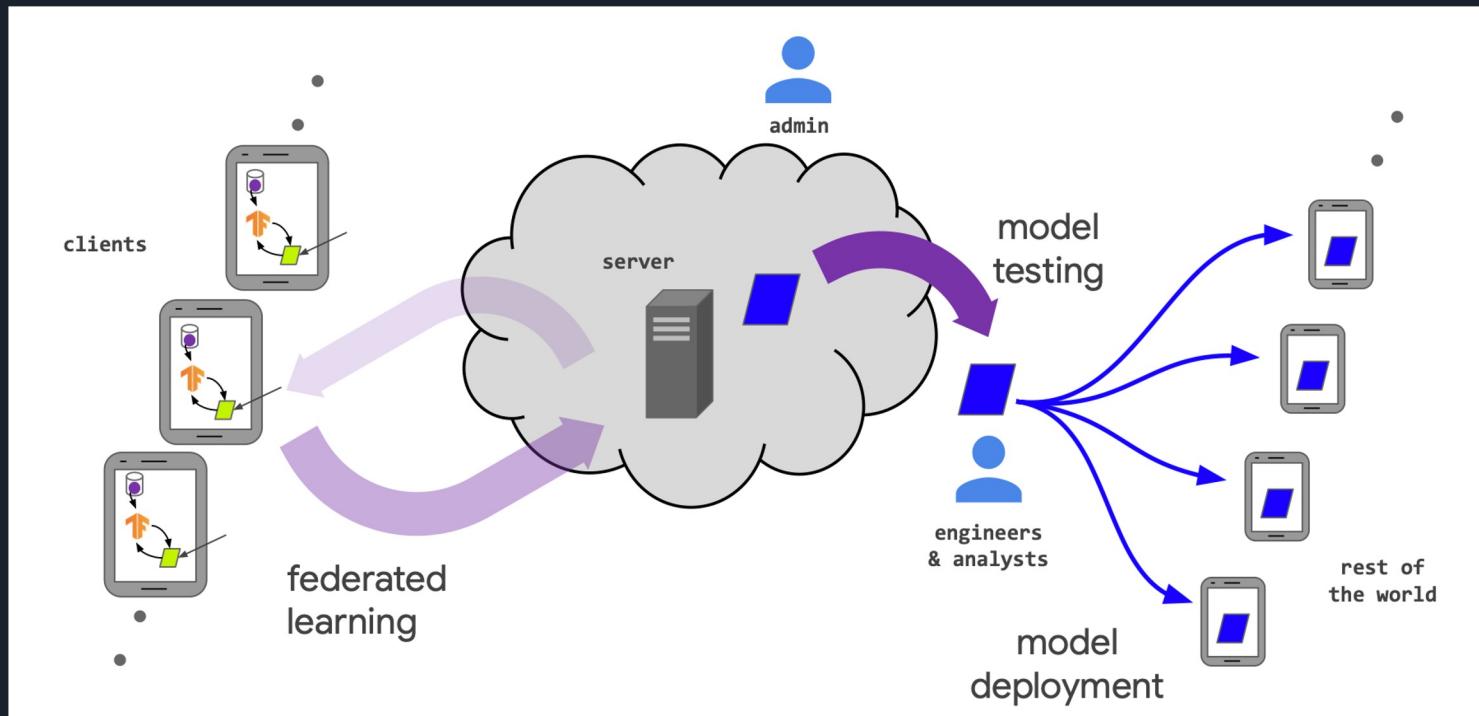
Hypotheses

- Differential privacy could affect the model's performance
- Centralized Machine Learning and Federated Learning models could produce similar accuracy, but federated learning may require more training cycles

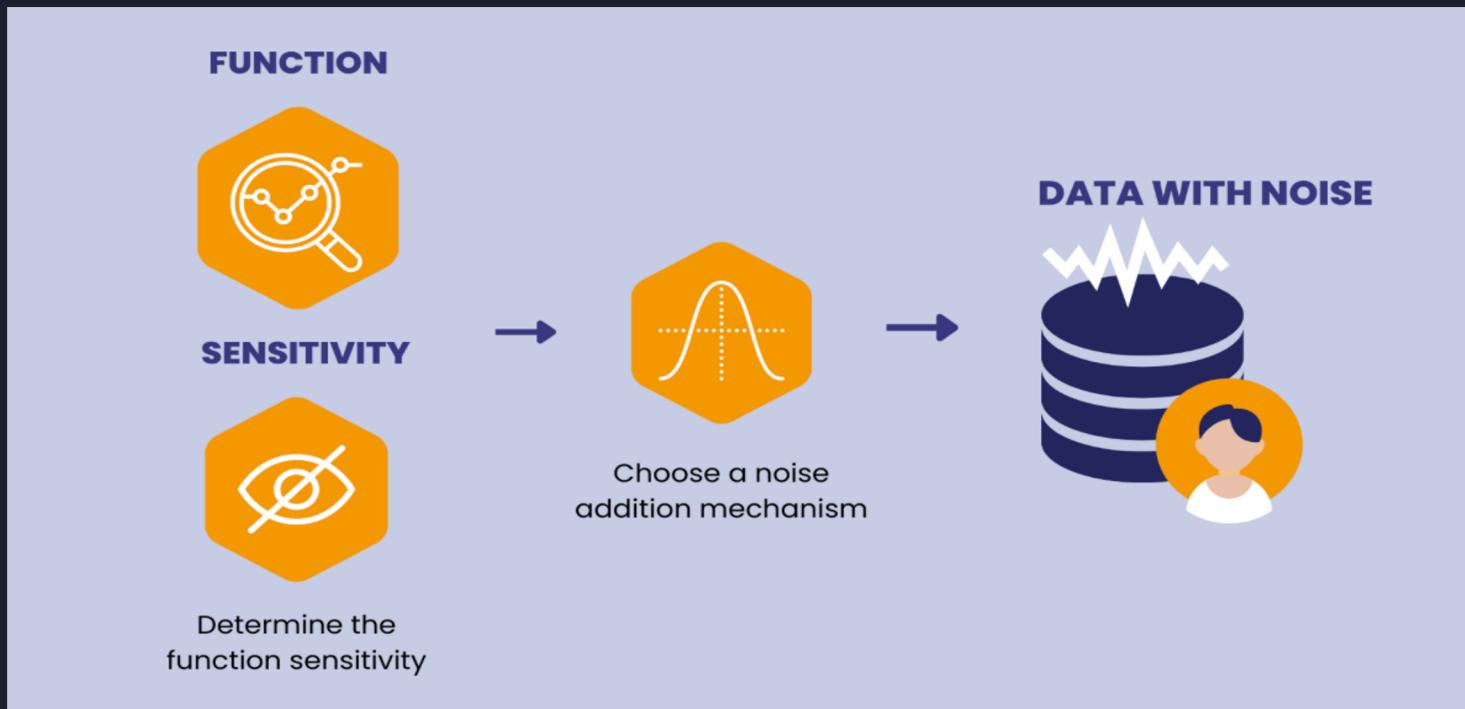
RELATED WORK

- The hospitals' updates are susceptible to information leakage about the patients' data due to model over-fitting to training data (Carlini et al. 2019).
- Differential privacy (DP) (Dwork and Roth 2014) limits an adversary's certainty in inferring a patient's presence in the training dataset.
- Before optimizing the DNN, Gaussian random noise is added to the computed gradients on the patients' data to achieve differentially-private stochastic gradient descent (DPSGD) (Abadi et al. 2016).
- Differential privacy in combination with other techniques such as secure aggregation, homomorphic encryption can establish a better privacy-utility trade-off (Malekzadeh, 2019)

WHAT IS FEDERATED LEARNING?



WHAT IS DIFFERENTIAL PRIVACY?



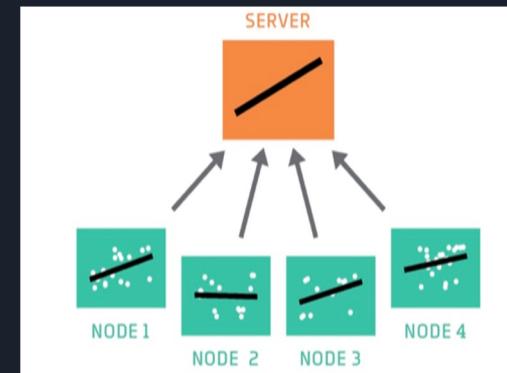
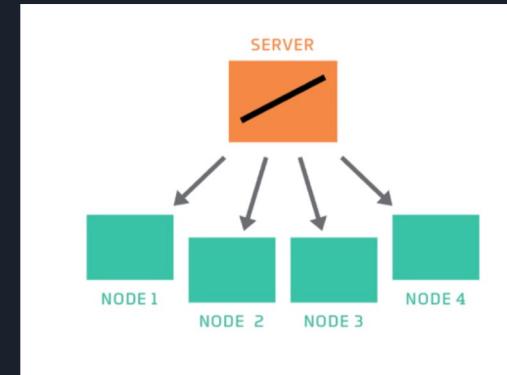
ABOUT THE DATASET

Retinal optical coherence tomography (OCT) is an imaging technique used to capture high-resolution cross sections of the retinas of living patients.

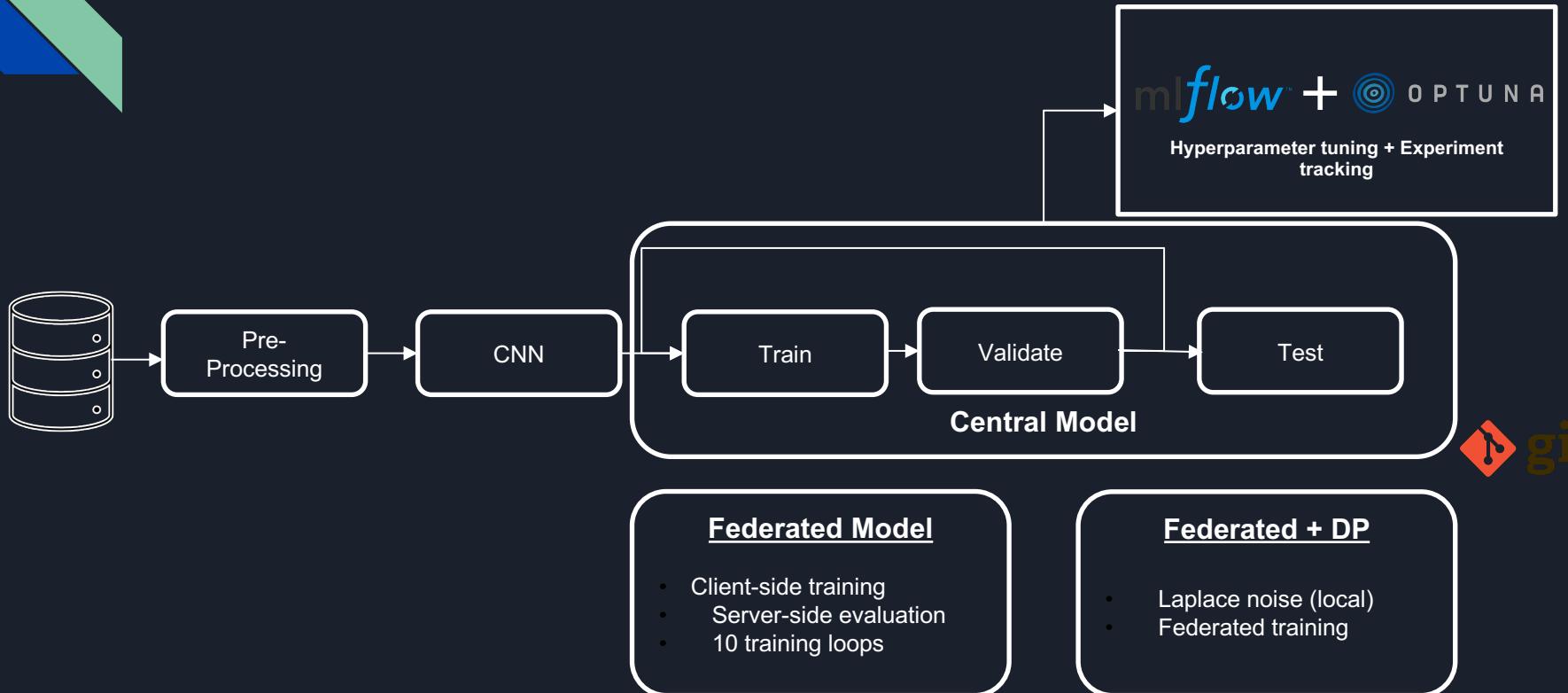


ALGORITHM: FEDERATED AVERAGING

- 1) The server has on it a machine learning untrained model.
- 2) Server sends a copy of the model to the nodes/clients.
- 3) So the nodes acquire or load some training data. Each node has different training data.
- 4) Then each node incrementally trains the model.
- 5) Now each node sends a copy of their model back to the server.
- 6) The server now has four models and it takes the average. And that average model encapsulates information that was present on every node.
- 7) That was one round of federated learning.
- 8) And then we repeat, i.e. the newly improved model is sent back to the nodes, and they train some more, and send further improved models back.



WORKFLOW



MODEL STRUCTURE

- Hyperparameter Tuning (Optuna + Mlflow)

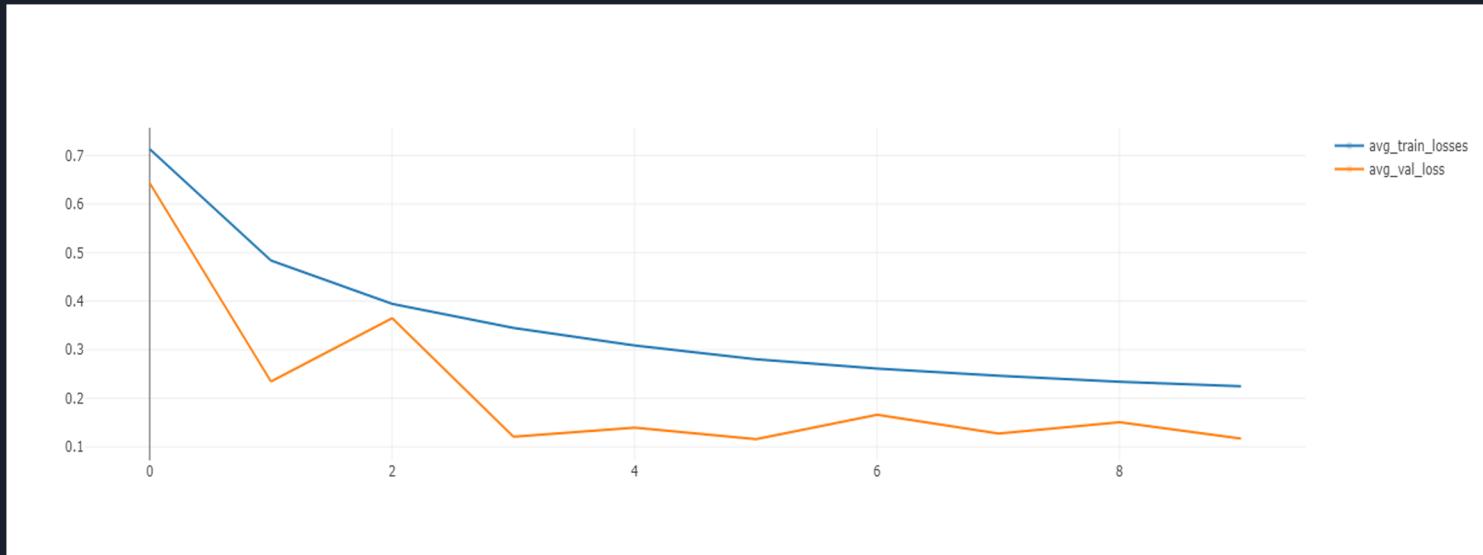
- Learning Rate = 0.03134
- Momentum = 0.9
- Dropout = 0.2
- Optimizer = Stochastic Gradient Descent (SGD)

```
Retina_Model(  
    (conv1): Conv2d(3, 32, kernel_size=(3, 3), stride=(1, 1))  
    (conv2): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1))  
    (dropout1): Dropout2d(p=0.2, inplace=False)  
    (dropout2): Dropout2d(p=0.2, inplace=False)  
    (fc1): Linear(in_features=12544, out_features=128, bias=True)  
    (fc2): Linear(in_features=128, out_features=4, bias=True)  
    (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1  
)
```

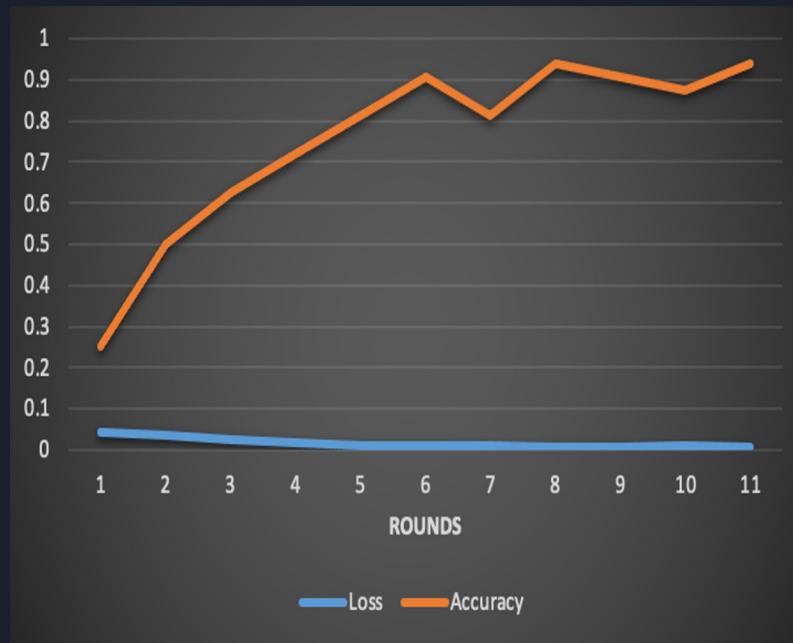
- Loss: Cross Entropy
- Num_Clients: 4 (Number of clients in Federated Learning model)
- Regularization: Dropout and Early Stopping

CENTRALIZED MODEL

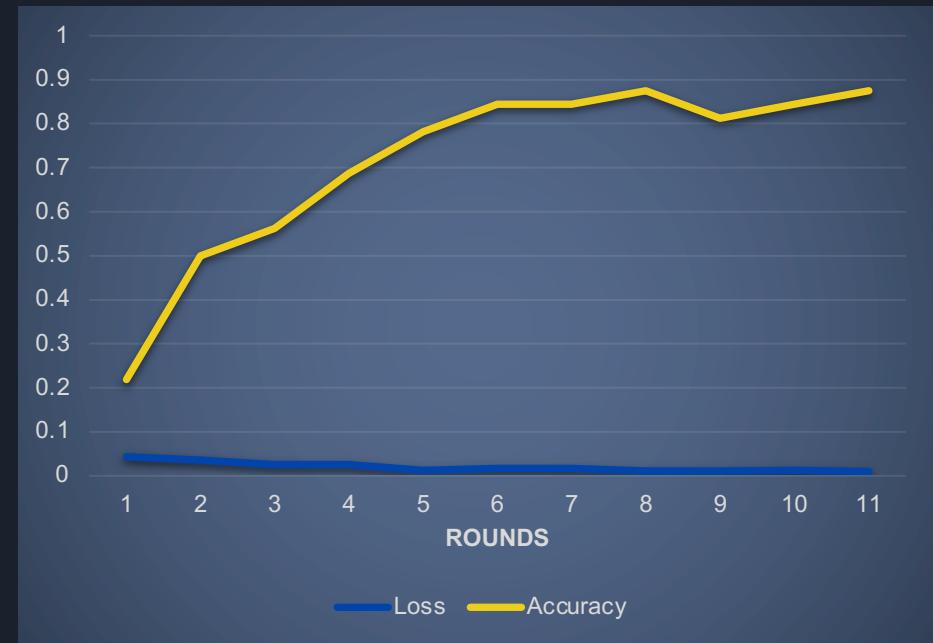
Training and Validation losses



LOSS & ACCURACY ACROSS FEDERATED LEARNING ROUNDS



FEDERATED



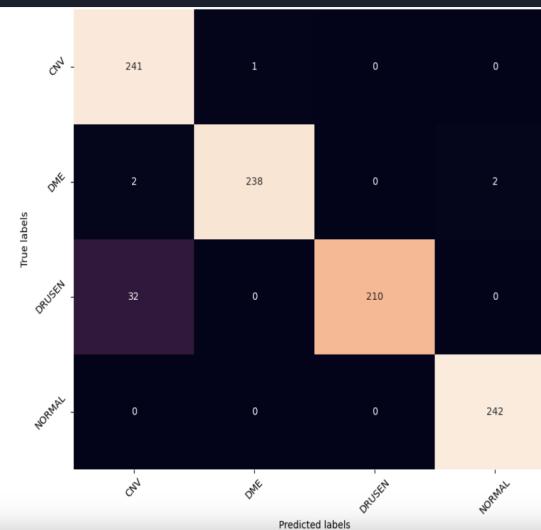
FEDERATED + DIFFERENTIAL PRIVACY

MODEL RESULTS ON THE TEST SET

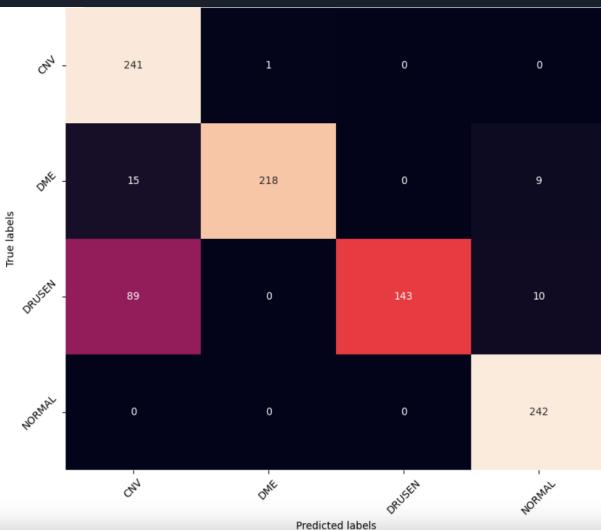
	Centralized		Federated		Federated + Differential Privacy	
	Precision	Recall	Precision	Recall	Precision	Recall
CNV	0.876	0.996	0.699	0.996	0.824	0.988
DME	0.996	0.983	0.995	0.901	0.982	0.897
DRUSEN	1.000	0.868	1.000	0.591	1.000	0.645
NORMAL	0.992	1.000	0.927	1.000	0.804	1.000
Accuracy	0.962	0.962	0.872	0.872	0.882	0.882
Macro Average	0.966	0.962	0.905	0.872	0.903	0.882
Weighted Average	0.966	0.962	0.905	0.872	0.903	0.882

CONFUSION MATRIX

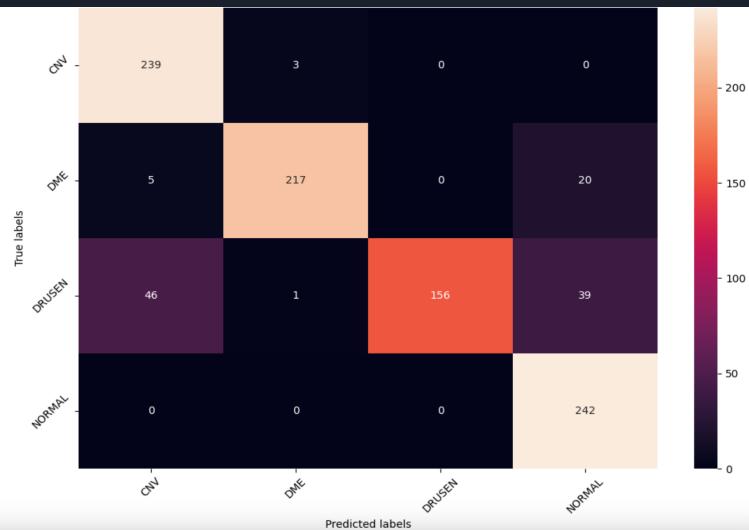
CENTRALIZED



FEDERATED



FEDERATED + DIFFERENTIAL PRIVACY



CONCLUSION

- With differential privacy being introduced, there exists privacy-accuracy tradeoff. As we guarantee more privacy, accuracy goes down and vice-versa. This can be seen from the validation set results:
 - Centralized Machine Learning model: 94%
 - Federated Learning model: 93.5%
 - Federated Learning with Differential Privacy model: 87.5%
- By changing sensitivity and epsilon, the amount of noise to add to the dataset can be adjusted to achieve desired levels of privacy and accuracy.
- Deployment: The Federated learning model can be used to detect eye diseases; training the model on datasets from multiple hospitals can be more accurate compared to a single hospital; integration into mobile apps; cloud-based API integrated with Electronic Health Record (EHR) systems etc.

FUTURE DIRECTION

- Problems with Linear Convergence Rates
 - Low bandwidth channels
 - Limited battery power
 - Client drift
 - Heterogeneity of the system
- Fed-LIN algorithm to facilitate convergence (Mitra, 2021)
- Differential Privacy Stochastic Gradient Descent (DP-SGD)
- Hyperparameter tuning in Federated Learning



Presidential Speeches



TEXT CLASSIFICATION

MARILIE BRETON, MARIA BIANCA GIOADA, SAMIRA YAZDANPOUR

Task



Supervised learning

Create a model that can classify the political parties in **Democratic** or **Republican** based on an American politician's speech

Overview

1. Data exploration
2. Statistical analysis
3. Text preprocessing
4. Model training
5. Prediction & Results



Dataset

Size : 988 documents

Number of variables: 7

- only using 2 : Party and Summary

presidential_speeches.csv (22.73 MB)

Detail Compact Column

7 of 7 columns ▾

Date	President	Party	Speech Title	Summary	Transcript	URL
1789-04-30	George Washington	Unaffiliated	First Inaugural Address	Washington calls on Congress to avoid local and party partisanship and encourages the adoption of a ...	Fellow Citizens of the Senate and the House of Representatives : Among the vicissitudes incident to l...	https://millercenter.org/the-presidency/presidential-speeches/april-30-1789-first-inaugural-address
1789-10-03	George Washington	Unaffiliated	Thanksgiving Proclamation	At the request of Congress, Washington establishes November 26th as a national holiday of thanks to ...	Whereas it is the duty of all Nations to acknowledge the providence of Almighty God, to obey his wil...	https://millercenter.org/the-presidency/presidential-speeches/october-3-1789-thanksgiving-proclamation

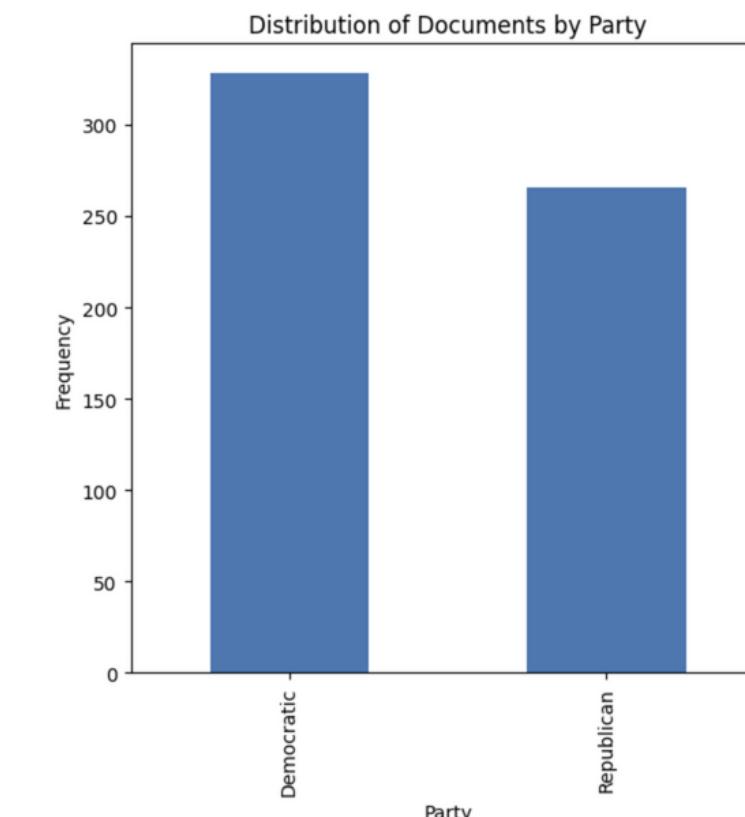
Label : binary variable

- democratic and republican

Number of transcripts per party (after cleaning)

Democratic : 328

Republican : 266

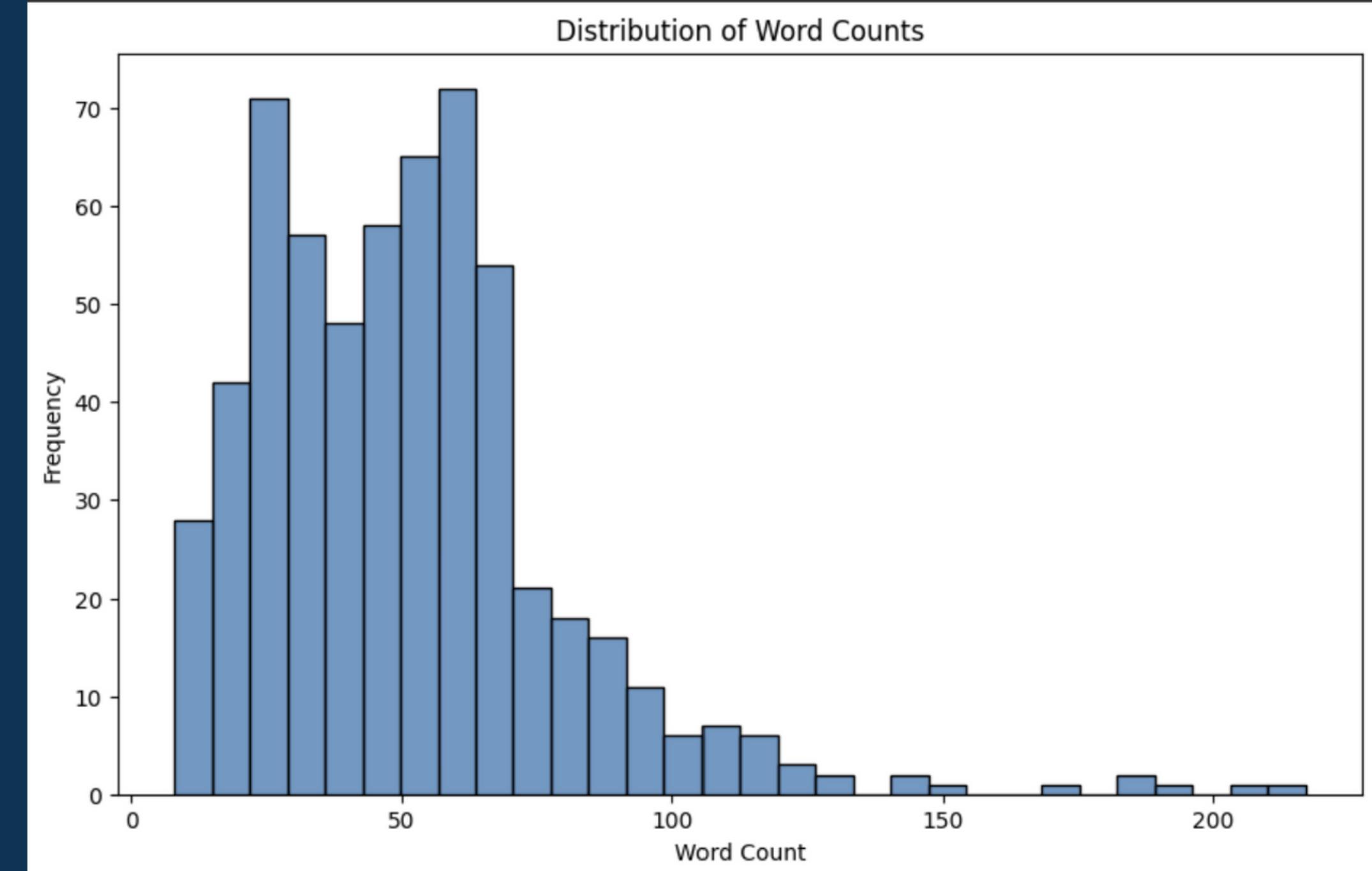


Statistical analysis

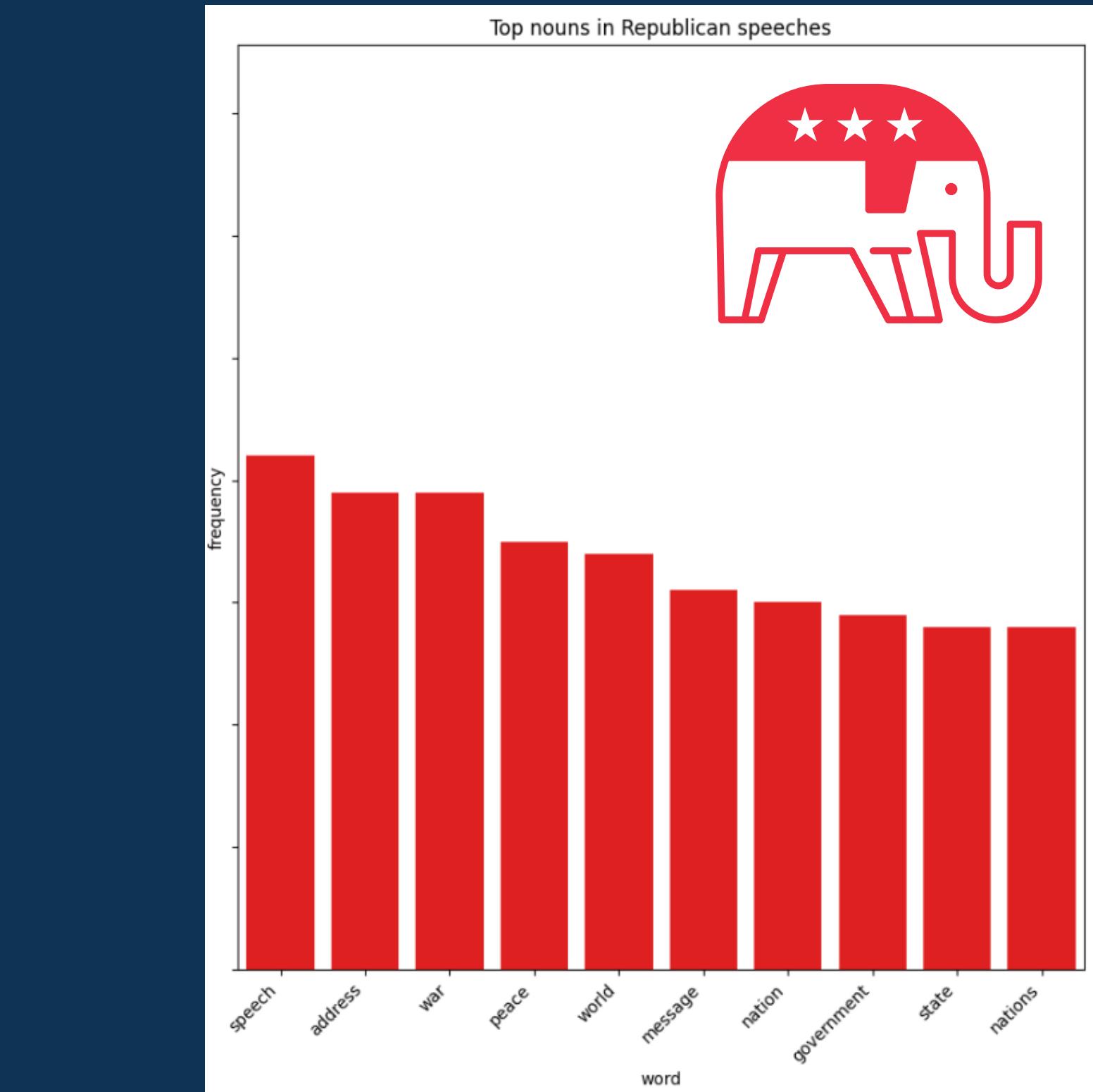
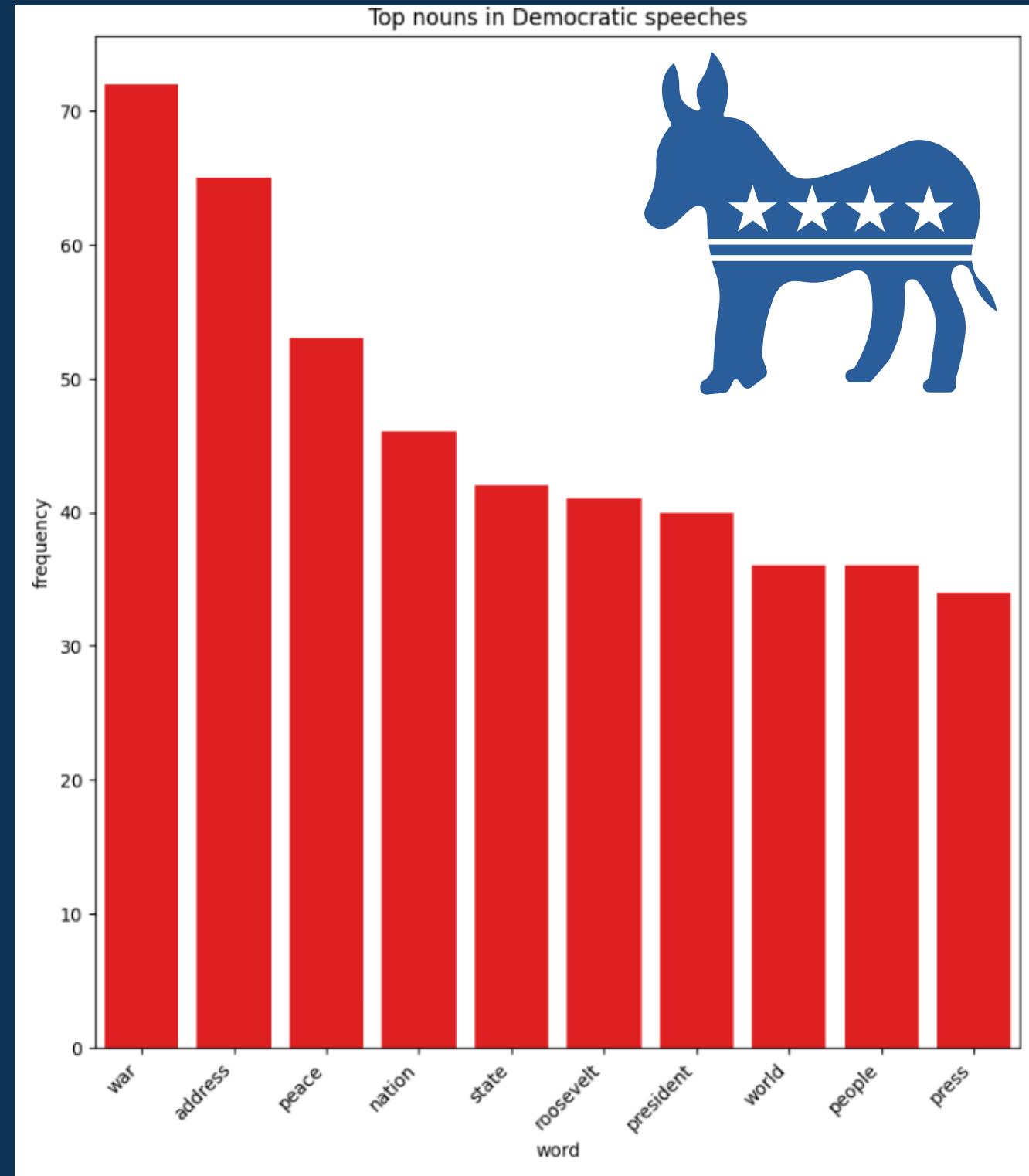
```
min          8.000000
25%        30.000000
50%        49.000000
75%        64.000000
max       217.000000
Name: word_count, dtype: float64

Distribution of documents by party:
Democratic    328
Republican    266
Name: Party, dtype: int64

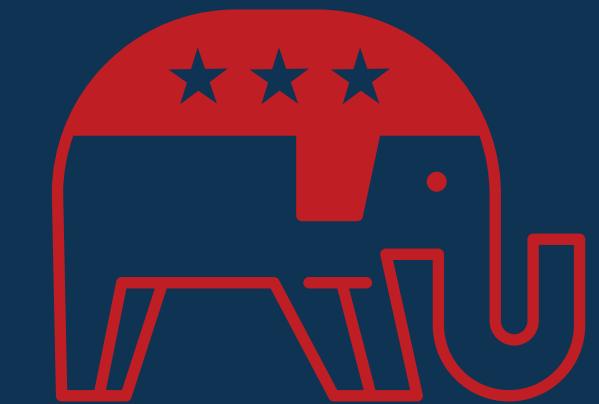
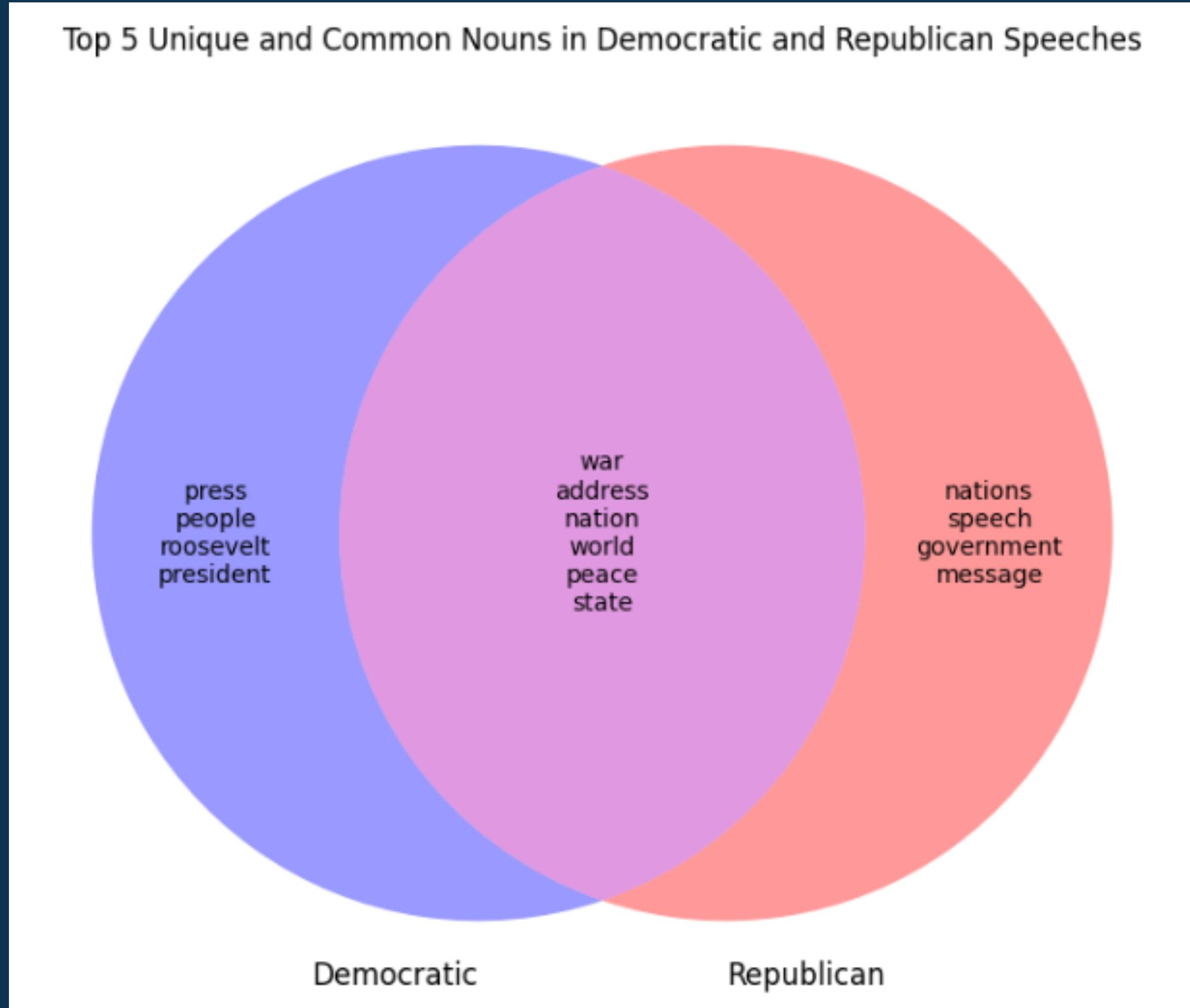
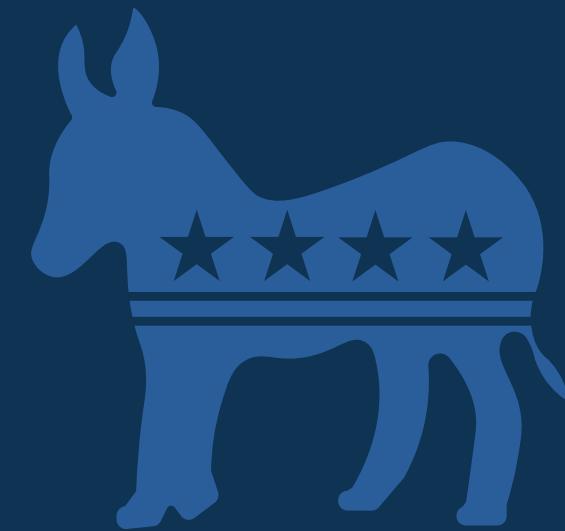
Average word count by party:
Party
Democratic    51.381098
Republican    50.646617
Name: word_count, dtype: float64
```



Frequency of words per party



Venn diagram



Approach

1

Text processing using TFIDF

2

Test simple & complex models

3

Compare results with baseline models



Text Processing

1. Remove missing values & special characters
2. Split data in train, test & validation
3. Create a vocabulary & TFIDF matrix
 - matrix dimension : 594 x 4643



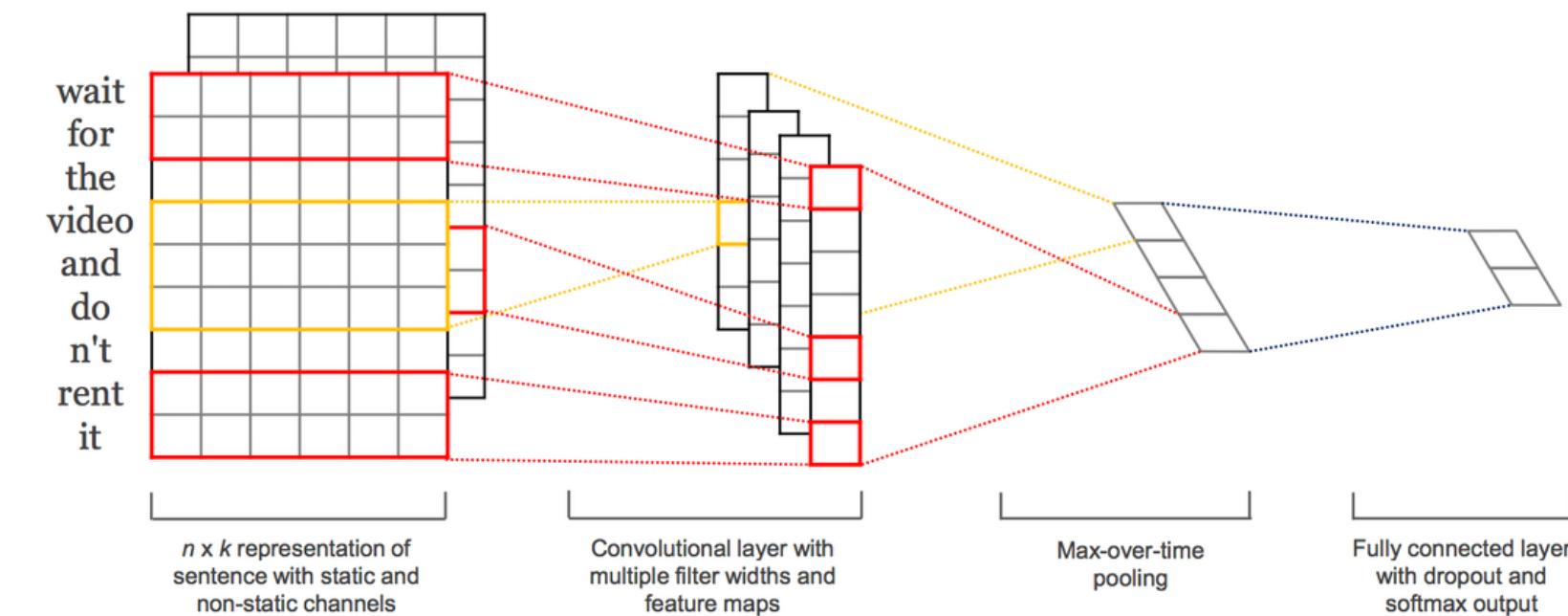
Models trained

Naive Bayes

SVM

Logistic Regression

CNN (4 layers)
convolutional,
activation(ReLU), max
pooling & fully connected



Pre-trained BERT for Sentiment Analysis

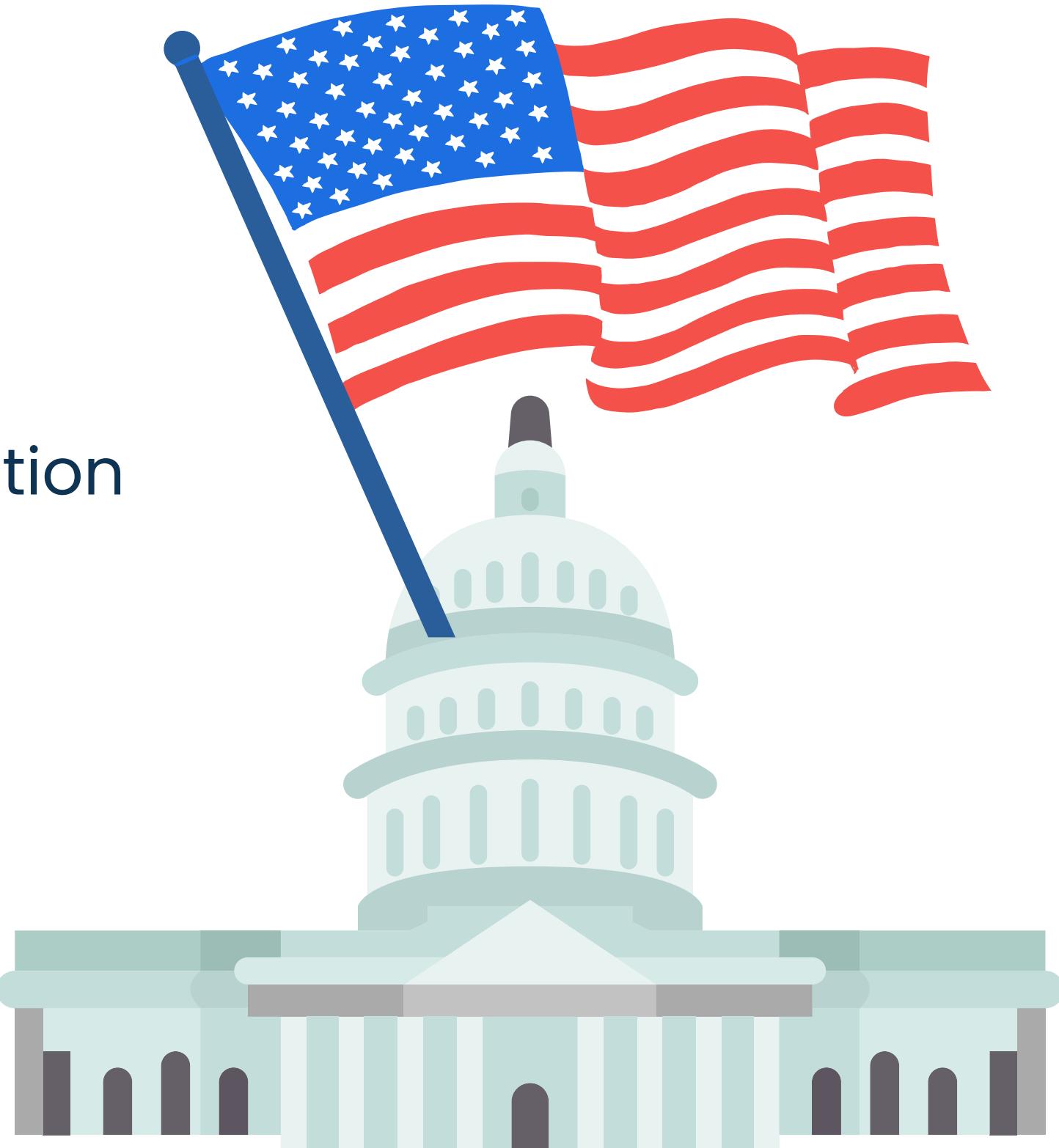
Evaluation setup

Baseline model: tokenize + model

Data split : 70% train, 15% test and 15% validation

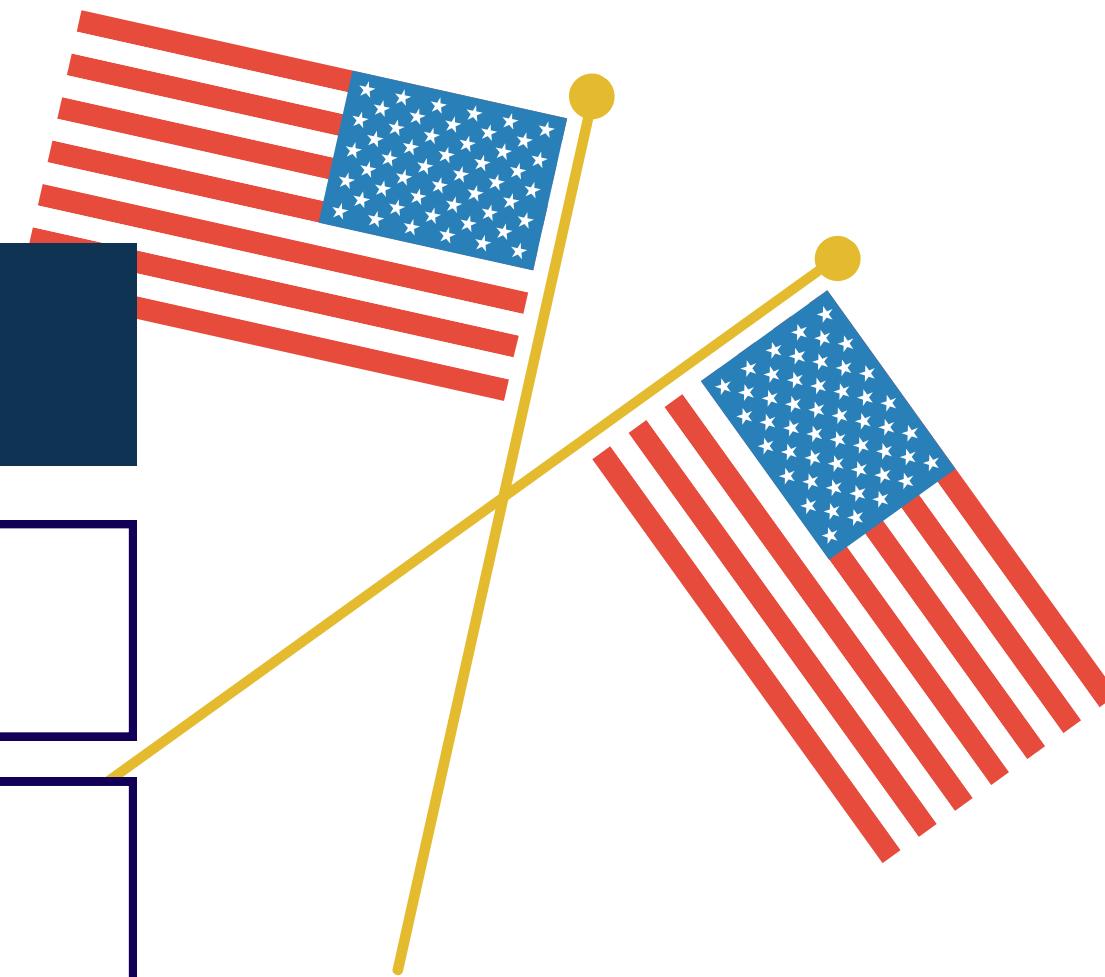
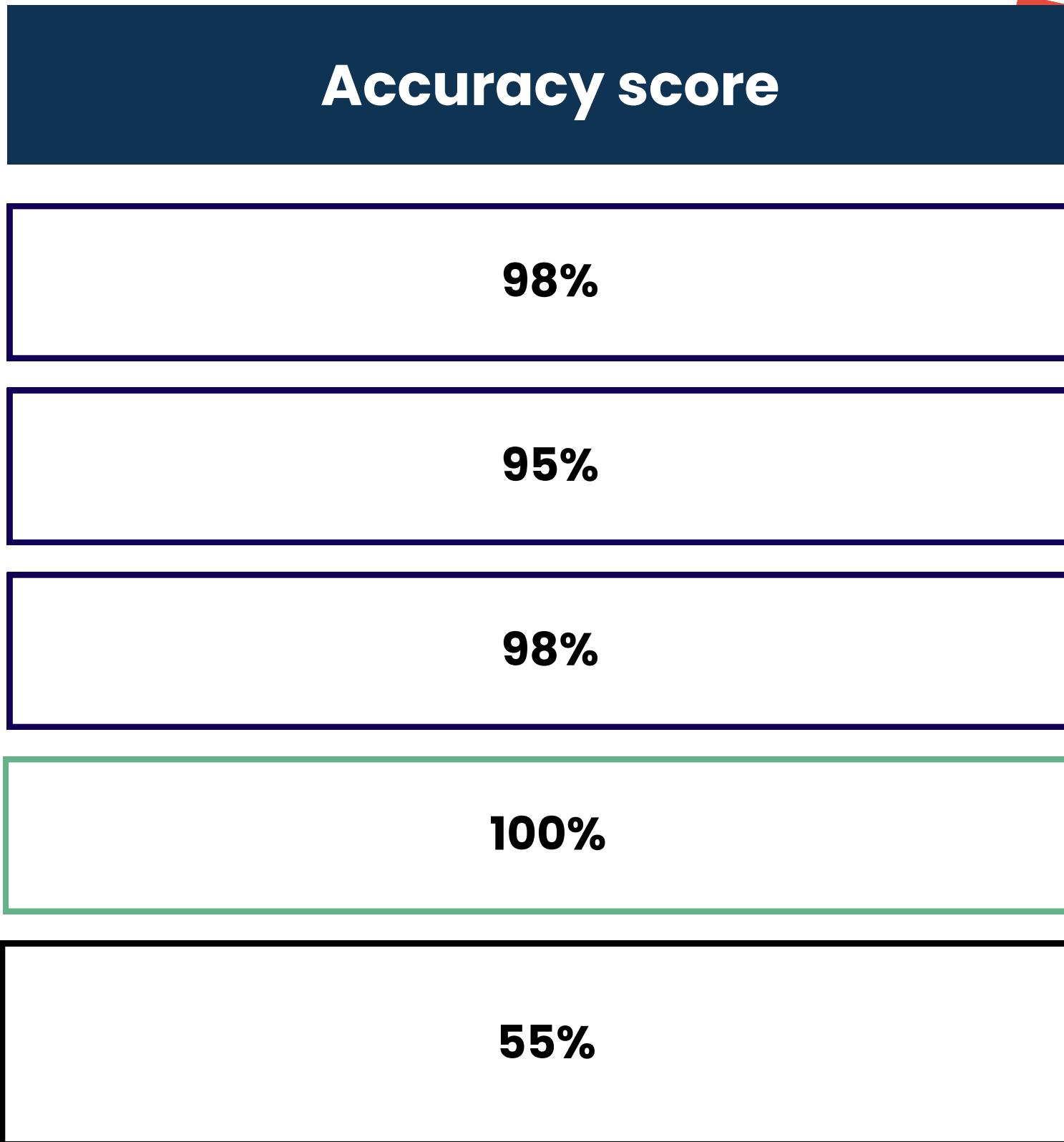
Hyperparameter tuning

- using research grid

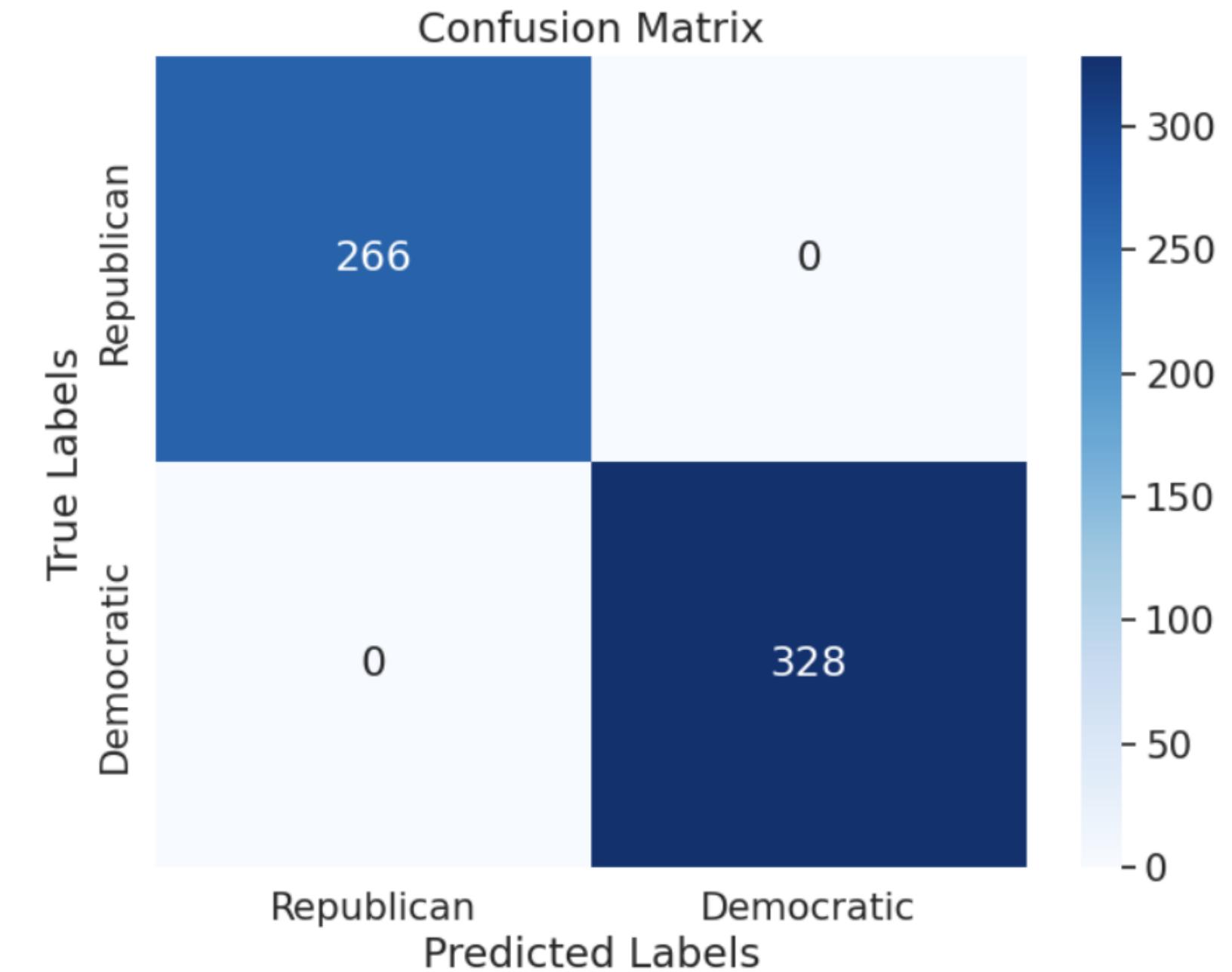
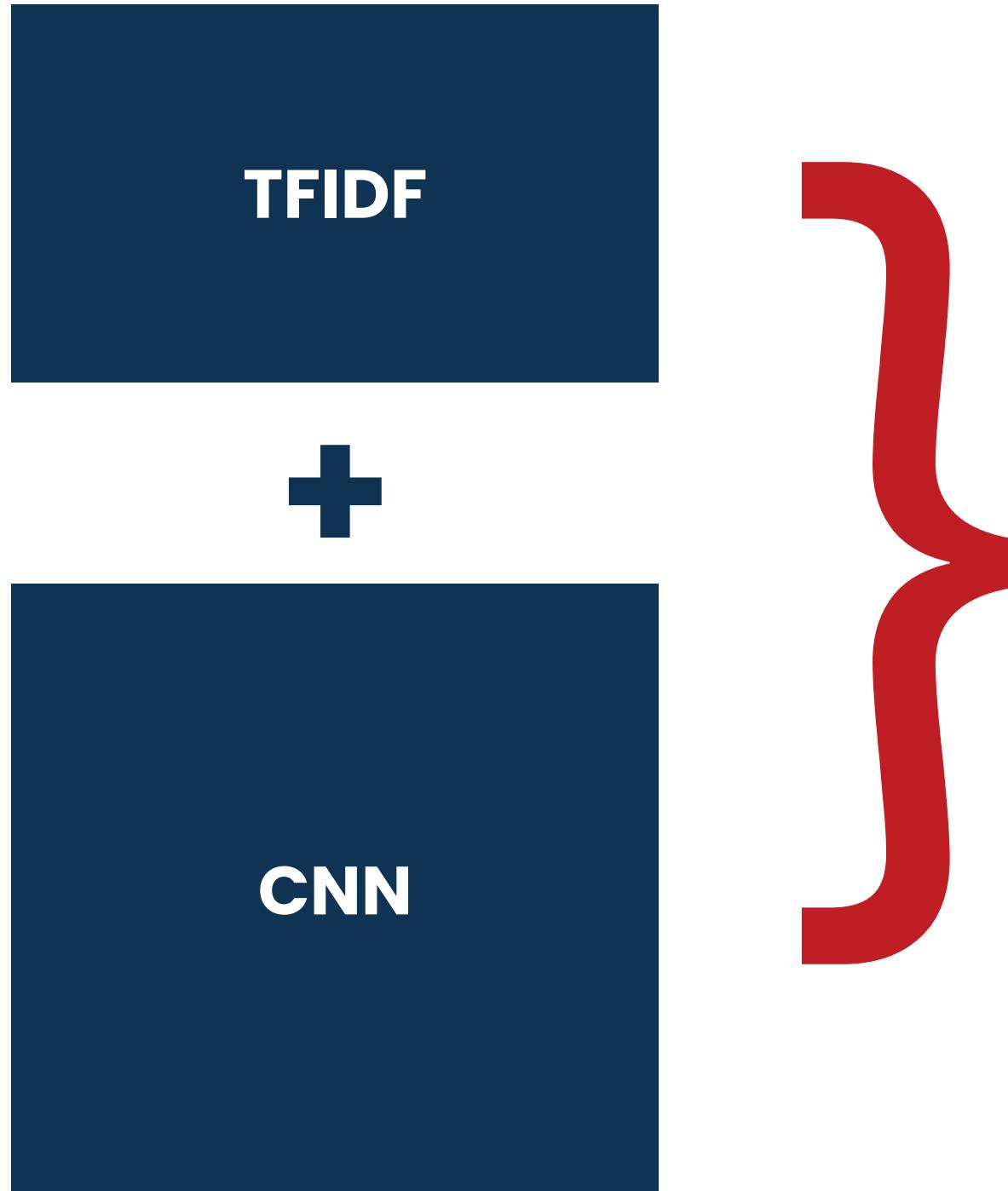


Evaluation

SVM
Naive Bayes
Logistic Reg
CNN
Pre-trained BERT for
Sentiment Analysis



Chosen model



What's next?

Improvements

- Test with transcript data instead of summary to increase the complexity of the classification task

Extend the work

- Adapt the model to other types of documents that can be classified in 2 categories (e.g. fake news)





Questions?



THANK YOU

MARILIE BRETON, MARIA BIANCA GIOADA, SAMIRA YAZDANPOUR



PATRICK CIMON
&
AYMAN BEN MANSOUR
MATH 80629A

Is stock return prediction
still impossible?

AGENDA



PATRICK CIMON
&
AYMAN BEN MANSOUR
M A T H 8 0 6 2 9 A

01 Context and Task

02 Related Works

03 Datasets

04 Approach and Evaluation Setup

05 Results

06 Future Direction



Stock Return Prediction

Context

It is common knowledge amongst portfolio manager that the market shows at least a weak efficiency, which means that past stock prices and their fluctuations do not contain any information about future stock prices.

With recent advances in machine learning is it possible to predict the direction public stock's returns based on its past performance and macroeconomic environment?

Task

Predict next period stock returns' sign (0: negative, 1: positive) with a classification supervised learning algorithm for sequential data:

- Bayesion Structural Time Series

Related Works

- "Deep Learning for Stock Prediction Using Numerical and Textual Information" by Weiwei Shen, Yilin Wang, and Zexi Shen (2019)
 - "Predicting Stock Prices with a Feature Fusion LSTM-CNN Model" by Yujun Zhang, Jiahang Li, and Hongyu Liang (2019)
 - "A Comparative Study of LSTM and GRU for Stock Price Prediction Using Historical Data" by Ganesh Bonde and Ritesh Kadmawala (2020)
 - "Deep Learning for Stock Market Prediction: A Comparative Study of Different Architectures" by Yiding Shen, Yan Yu, and Zhaozheng Yin (2019)
 - "Stock Price Prediction using Deep Learning Techniques: A Survey" by Muhammad Asif et al. (2020)
 - "Stock Price Prediction Using Attention-based Deep Neural Networks" by I. Kavitha and G. Uma (2020)
 - "Stock Price Prediction using Multi-Modal Deep Learning Approach" by Sandeep Kumar et al. (2019)
-

Key Insights:

- GRUs generally outperform LSTM and CNN.
- The attention head mechanism allows the model to focus on the most relevant features in the input data, potentially leading to better performance compared to models without attention mechanisms.
- Multi-Modal (numerical and textual) approaches show promising results for stock price prediction. The CNNs and LSTM networks, which may result in a more accurate prediction compared to using either method alone.



Dataset

Input Data

- Source : Yahoo Finance
- Train-Val date split: 2000-2017-2021
- Train-Test date split: 2000-2021-2023
- Training Stocks 'MSFT'
 - "D" : 5796 data points
 - "W": 1196 data points
 - "M": 276 data points

Macro-Economics Historical Data

- Source: fredapi (Federal Reserve Bank of St-Louis)
- Consumer_Confidence_Index: UMCSENT
 - Producer_Price_Index: PPIACO
 - Consumer_Price_Index: CPIAUCSL
 - Weekly_Claims_For_Unemployment_Insurance: ICSA
 - GDP: GDPC1
 - Retail_Sales_ex_autos: RSXFS
 - Industrial_Production_and_Capacity_Utilization: INDPRO
 - Durable_Goods_Orders: DGORDER
 - NBER_Recession_marker: USREC

Hyper-parameters

- Phi : ["D", "W", "M"]
- Lamda : Number of lagged return
- Stock
- Date splits

Labels

- Binary(Ret. t+1) :
- 0 : Negative return
 - 1 : Positive return





Approach and Evaluation Setup

Model

Bayesian Structural Time Series

The model consists of three main components:

1. Kalman filter. The technique for time series decomposition.
2. Spike-and-slab method. In this step, the most important regression predictors are selected.
3. Bayesian model averaging. Combining the results and prediction calculation.

Hyper- Param Tunning

- Phi : "W" = Weekly returns
- Lamda : 2-3 lagged returns (weekly)

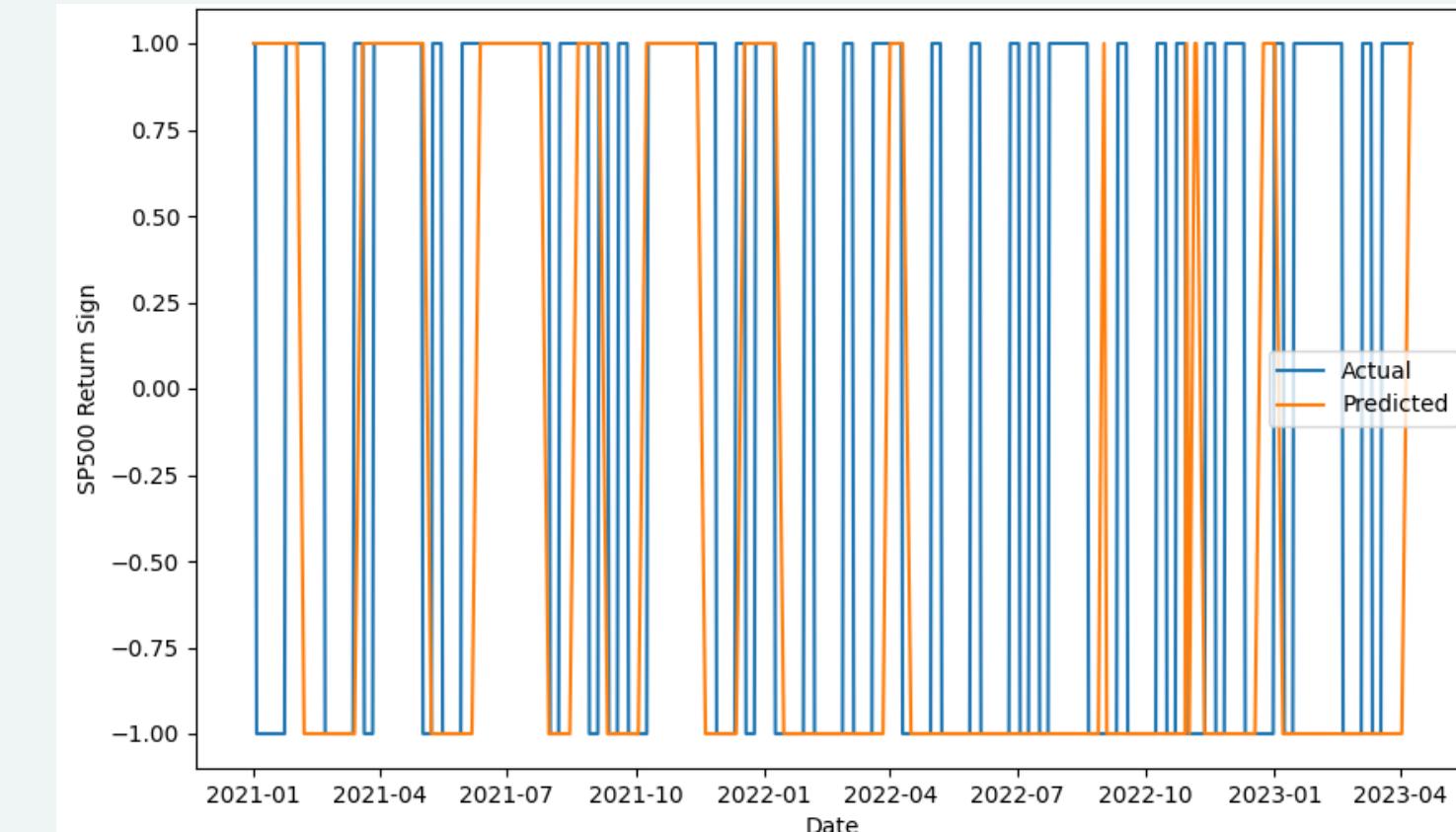
Date split

- Train-Val date split: 2000-2017-2021
- Train-Test date split: 2000-2021-2023

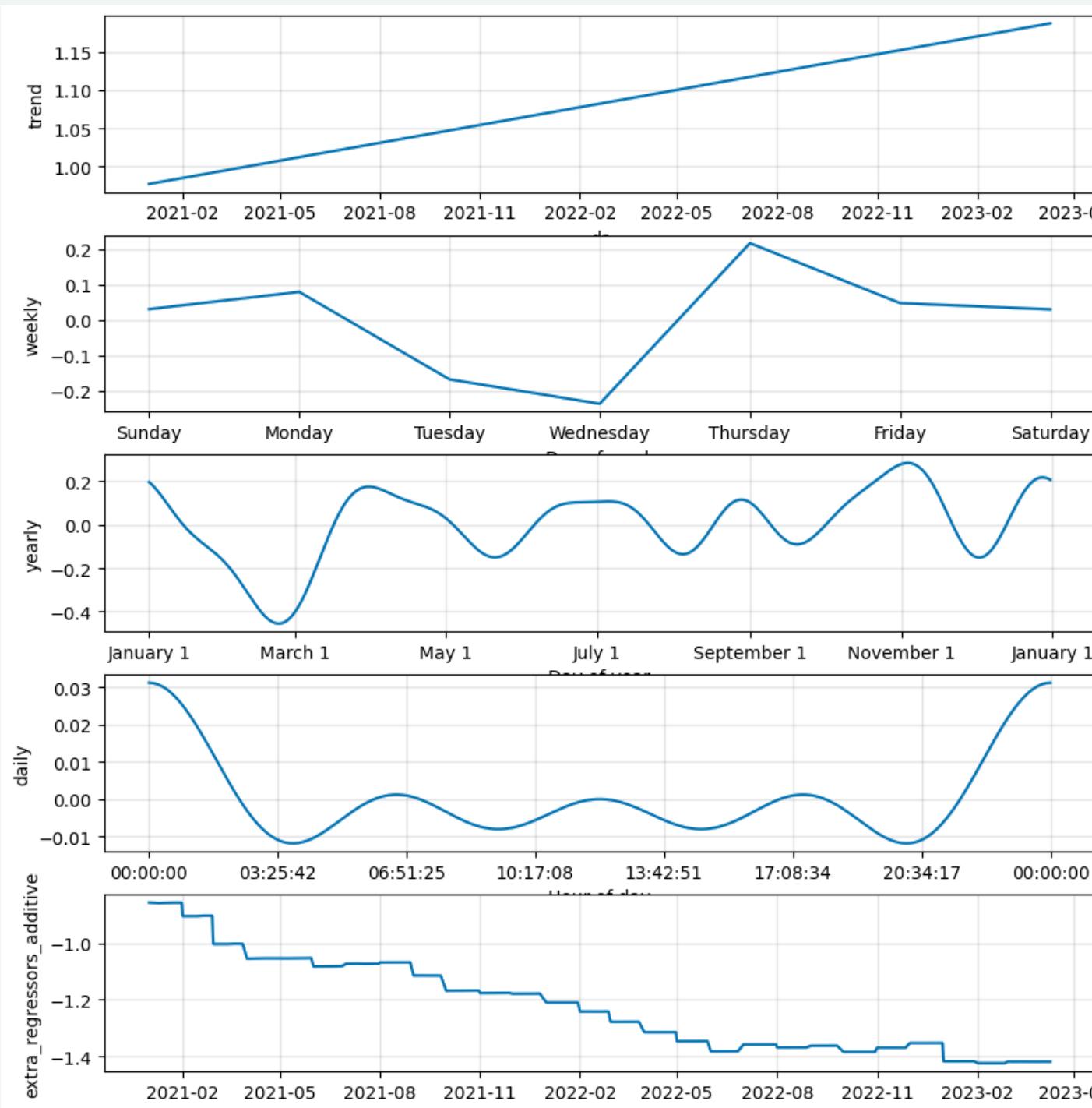
Results

ACCURACY
[51%-61%]

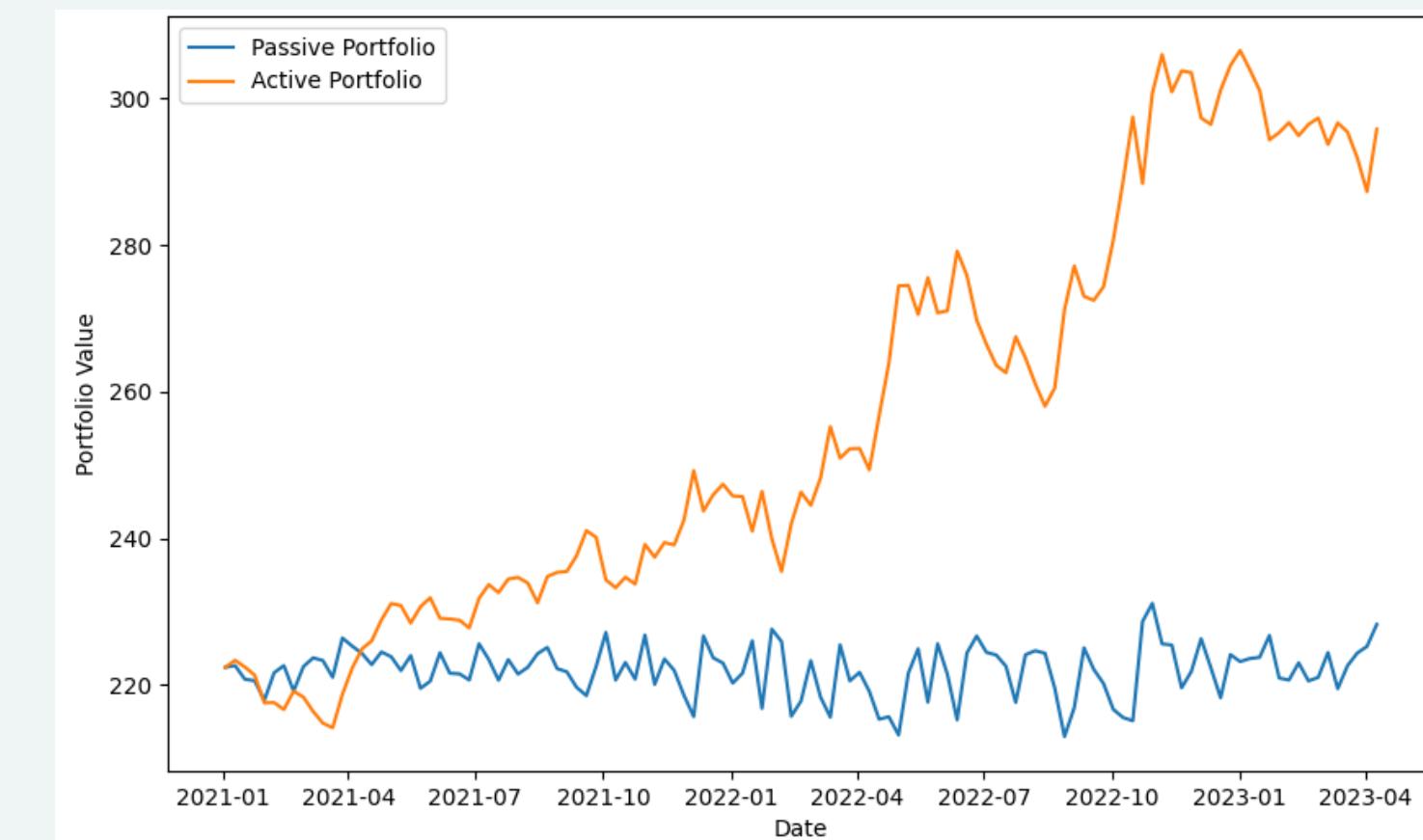
MSFT Returns' Sign Prediction



Trends and extra components



Active vs Passive Portfolio Evolution



Future Direction

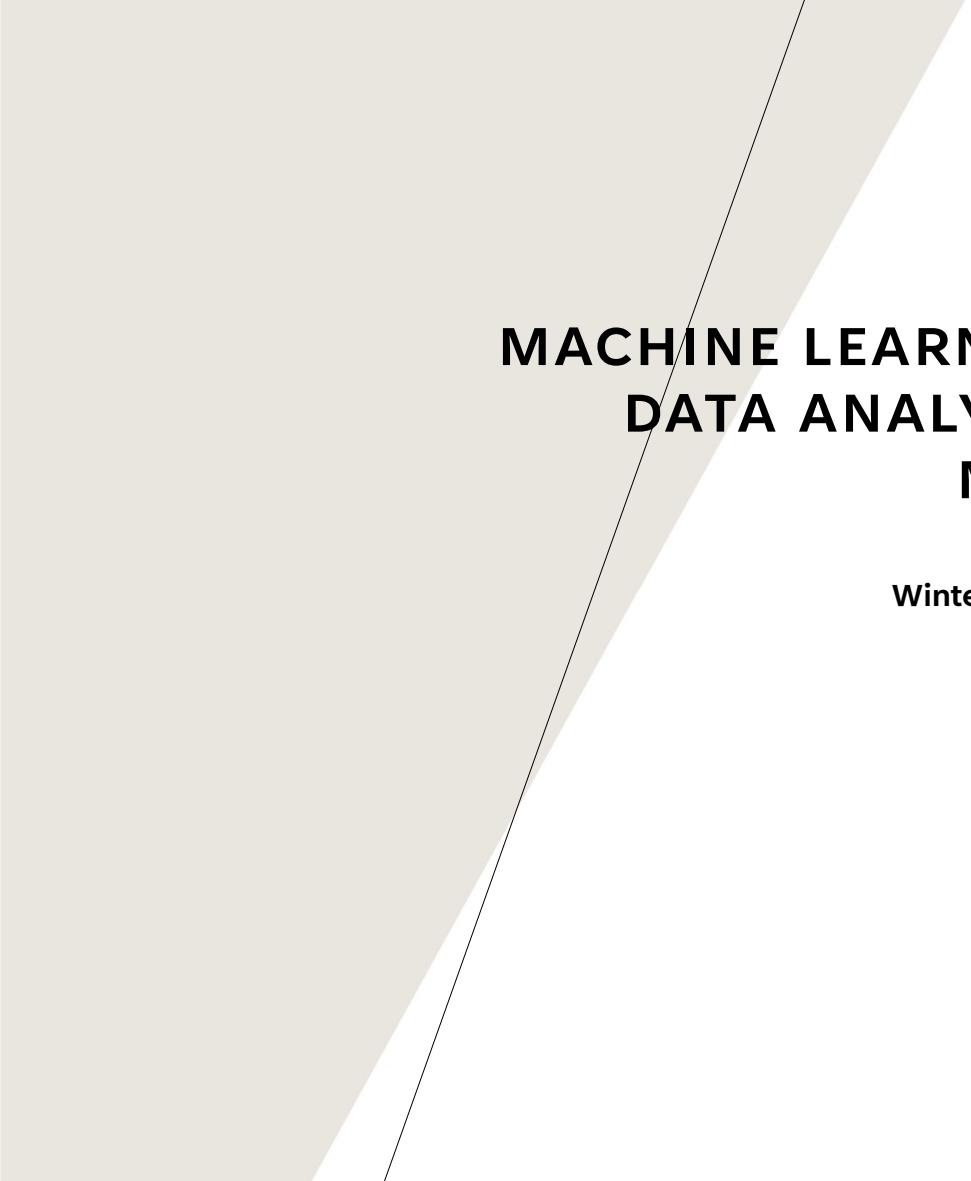
- The performance depends a lot on the hyperparameters (Data split, stocks, frequency, etc.)
- Implementation of portfolio prediction with multiple stocks instead of only one to benefit from the biased coin flip effect
- Data augmentation from multiple similar stocks historical data
- Model additions:
 - A transformer or residual connection and an attention head layer, to focus on the most relevant features in the input data
 - Recurrent neural network layer could be beneficial for the longterm dependencies

These changes may help make stock return prediction possible!

Thank you for listening!
Questions?



PATRICK CIMON
&
AYMAN BEN MANSOUR
M A T H 8 0 6 2 9 A



MACHINE LEARNING FOR LARGE SCALE DATA ANALYSIS AND DECISION MAKING

Winter 2023



MOVIE GENRE CLASSIFICATION

Houri Kechechian

Hadi Mohammadibajgiran

Gelareh Ghaderi

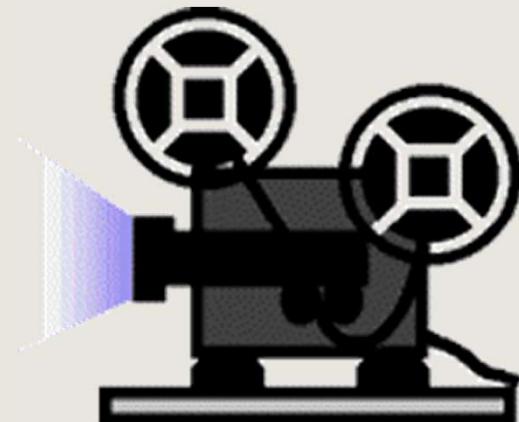
14th April 2023

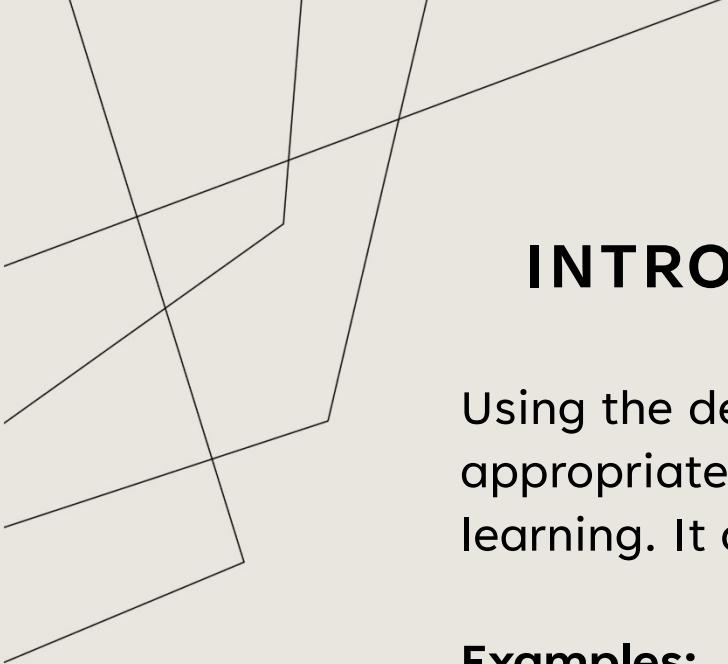


MOVIE GENRE CLASSIFICATION

AGENDA

- Introduction
- Movie Genre Dataset and task
- Literature review
- Our Approach
- Models of Machine learning
- Results and Model selection
- Future Direction





INTRODUCTION

Using the description of a movie to classify it into the appropriate genre is one of the interesting tasks in machine learning. It can also help businesses to achieve goals.

Examples:

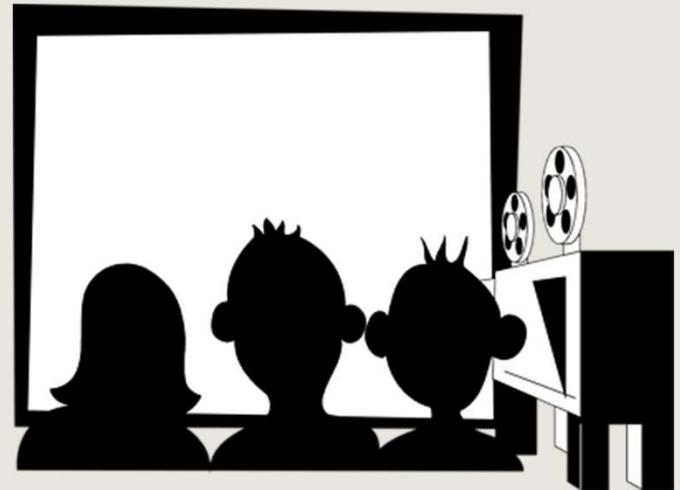
- Revenue generation
- Compliance
- Brand Reputation

TASK

- Classification (Supervised Learning)

APPLICATIONS

- Personalization
- Marketing
- Trend analysis



DATA SET

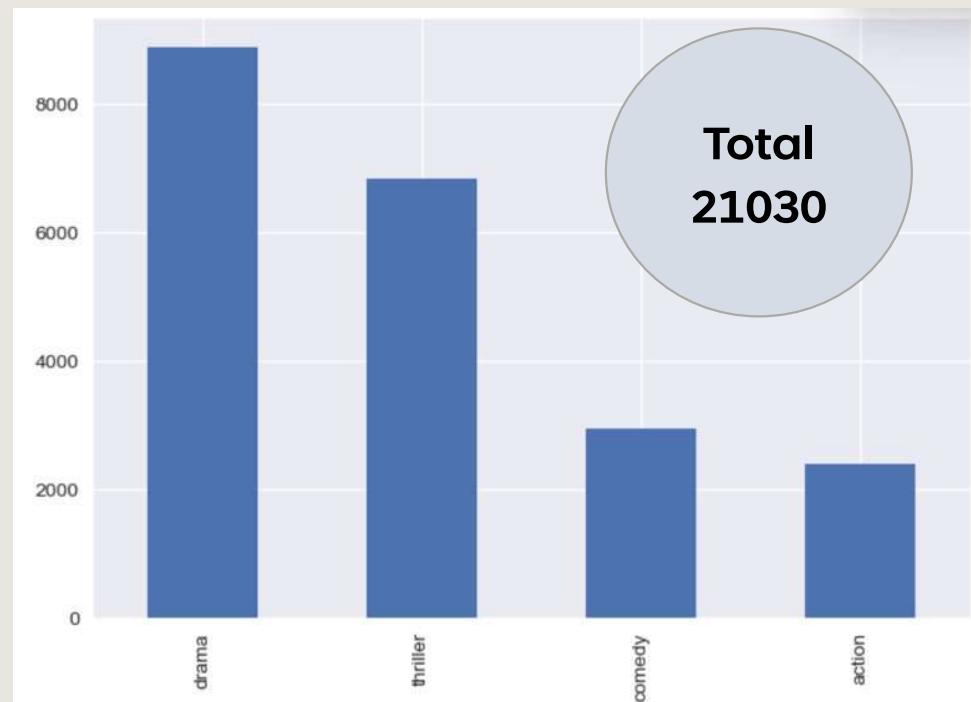
TEXT DATA

- Movie Trailer Script

CLASSES

- Action
- Drama
- Comedy
- Thriller

20XX



WORD CLOUD



LITERATURE REVIEW

Author(s)	Dates	Methods	Subject
Mert Ertugrul Pinar Karagoz	2018	Applied Bi-directional LSTM, and it performed better than baseline models (LR and Basic RNN)	Movie genre classification from plot summaries of movies
Nikhil Kumar Rajput Bhavya Ahuja Grover	2022	k-nearest neighbor (KNN) algorithm has the best performance	Movie genre classification scheme based on the movie's subtitles
SHERVIN MINAEE, Nal Kalchbrenner	2021	150 deep learning--based models for text classification	Analysis of the performance of different deep learning models on popular benchmarks,

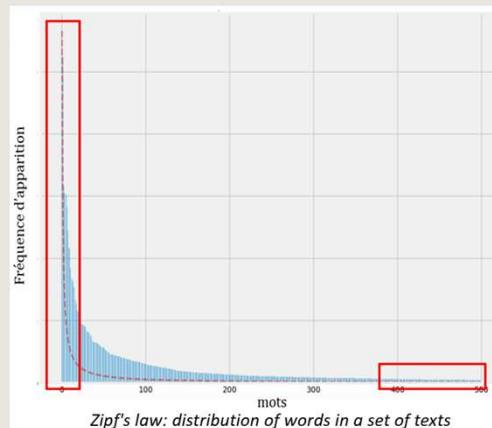
OUR APPROACH

=> WHY

1- Preprocessing data

- Lower Case
- Edit some expressions
- Stop word removal
- Lemmatization
- Embeddings (TF-IDF and Word2vec)

```
text = re.sub(r"n't", " not ", text)
text = re.sub(r'i'm", "i am ", text)
text = re.sub(r'\re", " are ", text)
text = re.sub(r'\d", " would ", text)
text = re.sub(r'\ll", " will ", text)
```



we use lemmatization for most applications unless we seek to strongly optimize recall.

we have large database and highly contextual and nuanced text



OUR APPROACH ==> WHY

2- Machine learning models

SVM

Data with high-dimensional feature space, and the classes are separable.

KNN

Data is non-linear and the number of classes is small.

Neural Network

Large data with high-dimensional feature space, and the classes are not easily separable

CNN

Large and complex sequential data

RNN

Large and complex sequential data

MACHINE LEARNING MODELS

SVM	KNN	Neural Network	RNN	CNN
TF-IDF C regularization kernel	TF-IDF n_neighbors weights	TF-IDF activation: relu hidden size max_iter, Optimizer: ADAM	Word Embeddings RNN structure: Embedding Layer Hidden layer (with and without LSTM) Fully Connected Layer Loss: Cross entropy Optimizer: ADAM	Word Embeddings CNN structure: Embedding Layer (convolution layer Activation function(ReLU) Max pooling)x2 Fully Connected Layer Loss: Cross entropy Optimizer: ADAM

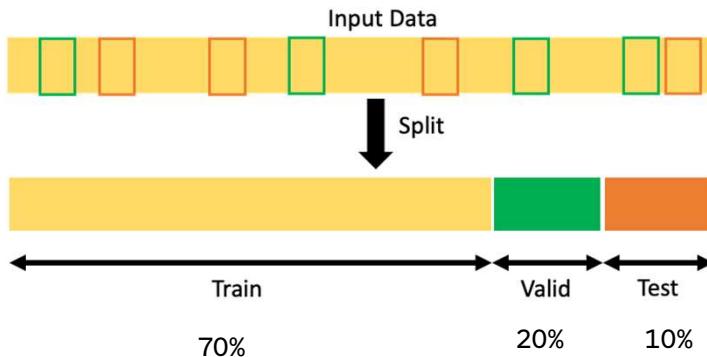
EVALUATION SETUP

Baseline Model:

Random Label Assignment

Splitting the data:

Split the data into Training (70%), Validation (20%) and Test (10 percent) sets.



Hyperparameter Tuning:

Performed hyperparameter tuning on the validation data set and the hyperparameters specific to each model

HYPERPARAMETER TUNING

SVM

Extensive trial and error (C, kernel)

KNN

Extensive trial and error (n_neighbors, weights)

Neural Network

GridSearchCV (hidden_layer_sizes, max_iteration, solver and activation function)

CNN

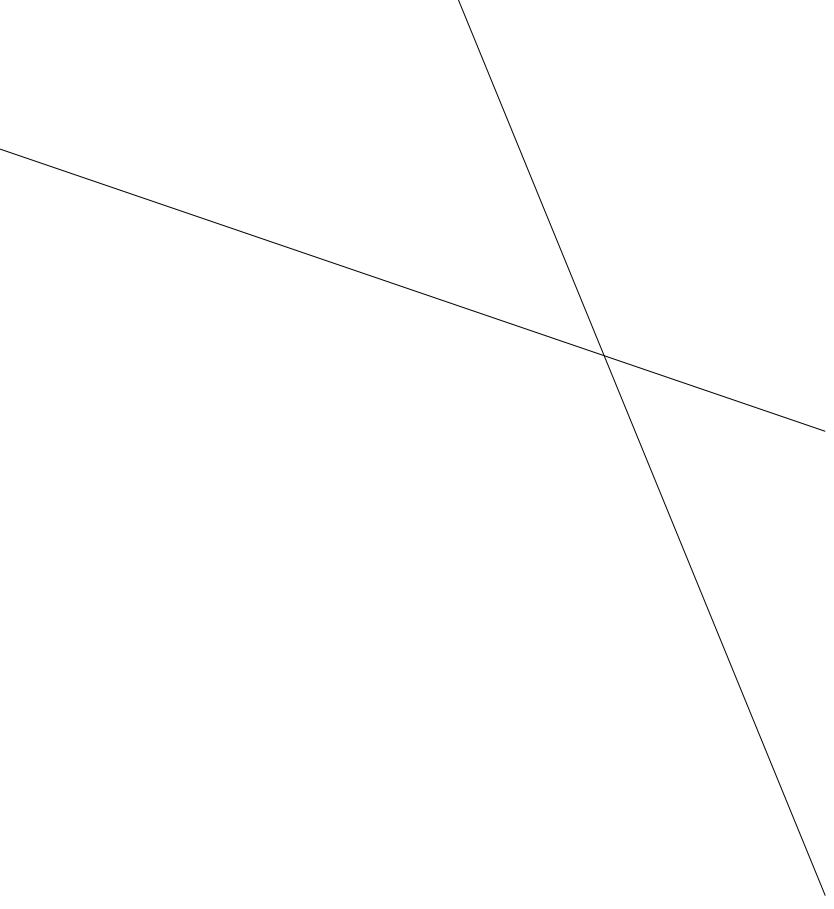
GridSearchCV (Learning rate, Filters, batch size, number of filters)

RNN

GridSearchCV (learning rate, epoch numbers)

MODEL COMPARISON AND SELECTION

	Baseline	SVM	KNN	Neural Networks	RNN	RNN (LSTM)	CNN
Accuracies	24.8%	72.14%	92.64%	94%	47.65%	86.55%	38.51%



FUTURE DIRECTION

- ❖ Using another Word Embedding to improve representation (Glove, BERT)
- ❖ Using a larger dataset for better model training
- ❖ Using another Tuning models (Dropout, Network architecture,...)
- ❖ Test another Models (Naïve base, Random Forest, ...)

REFERENCE

- [1] Ertugrul. A, Karagoz. P, 2018, Movie Genre Classification from Plot Summaries Using Bidirectional LSTM, IEEE, Laguna Hills, CA, USA
- [2] Rajput N. et al, 2022, A multi-label movie genre classification scheme based on the movie's subtitles, springer, 81, p. 32469–32490.
- [3] Minaee S. et al, 2021, Deep Learning--based Text Classification: A Comprehensive Review, ACM Computing Surveys, V (54), No. 62, p. 1–40.
- [4] database: [NLP\(movie genre classification\) | Kaggle](#)



THANK YOU!



CUSTOMER CHURN PREDICTION

MATH60629A.W2023
April 14, 2023

Javad Ghasempur Sis - Helen Ma - Jiahua Shang



Agenda

- ✓ Introduction
- ✓ Dataset
- ✓ Models
- ✓ Data preprocessing
- ✓ Results
- ✓ Conclusion & Future direction





What is churn?

"rate at which customers stop doing business with a company or cancel their subscription to a service"

Why is it important?

- ✓ Revenue Loss
- ✓ Customer Satisfaction
- ✓ Cost of acquisition > Retention cost
- ✓ Brand reputation
- ✓ Competitive advantage



Supervised Learning: Classification Task

Credit Card Customer Dataset



1

Target variable
"Attrition_Flag"

10K

observations on
customer data

14

numerical variables

5

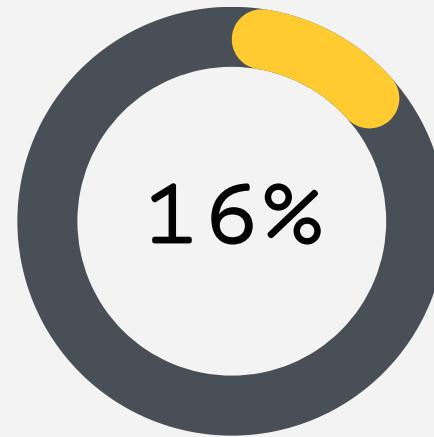
categorical variables

Variable	Description
Target	Attrition_Flag 1 if account is closed, 0 otherwise
Demographic	Gender Education_Level Marital_Status Income_Category Customer_Age Dependent_Count 2 levels {Male, Female} 7 levels {high school,..., graduate, other} 4 levels {Single, Married, Divorced, Unknown} 7 levels { <40K, 40K-60K, ...80K-120K, >120K} Customer's age (yrs) Number of dependents
Relationship with bank	Months_On_Book Total_Relationship_Count Period of relationship with bank Total # of products held by the customer
Activity	Months_Inactive_12_Mon Contacts_Count_12_Mon No. months inactive (last 12 mo) No. contacts (last 12 mo)
Credit card info	Credit_Limit Total_Revolving_Bal Avg_Open_To_Buy Card_category Credit limit on credit card Total revolving balance on credit card Open to buy credit line 4 levels {Blue, Silver, Gold, Platinum}
Transaction Info	Total_Amt_Chng_Q4_Q1 / Total_Ct_Chng_Q4_Q1 Total_Trans_Amt / Total_Trans_Ct Avg_Utilization_Ratio Change in transaction amount/count (last 12 mo) Total transaction amount/count (last 12 mo) Average card utilization ratio

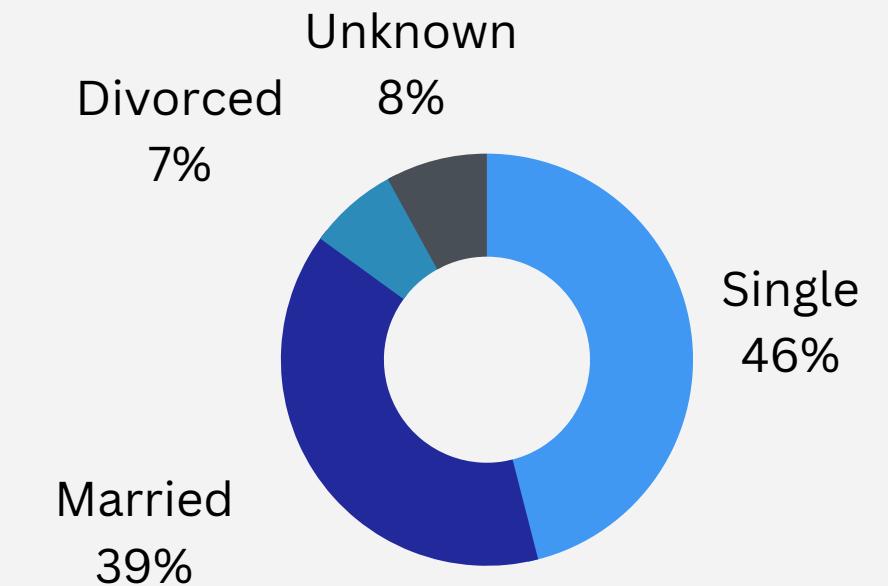
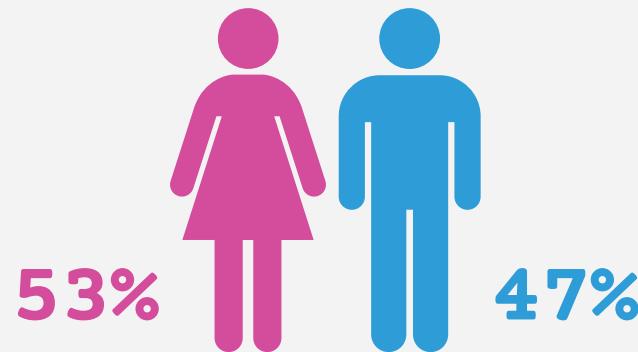
Data exploration

missing values
No missing values

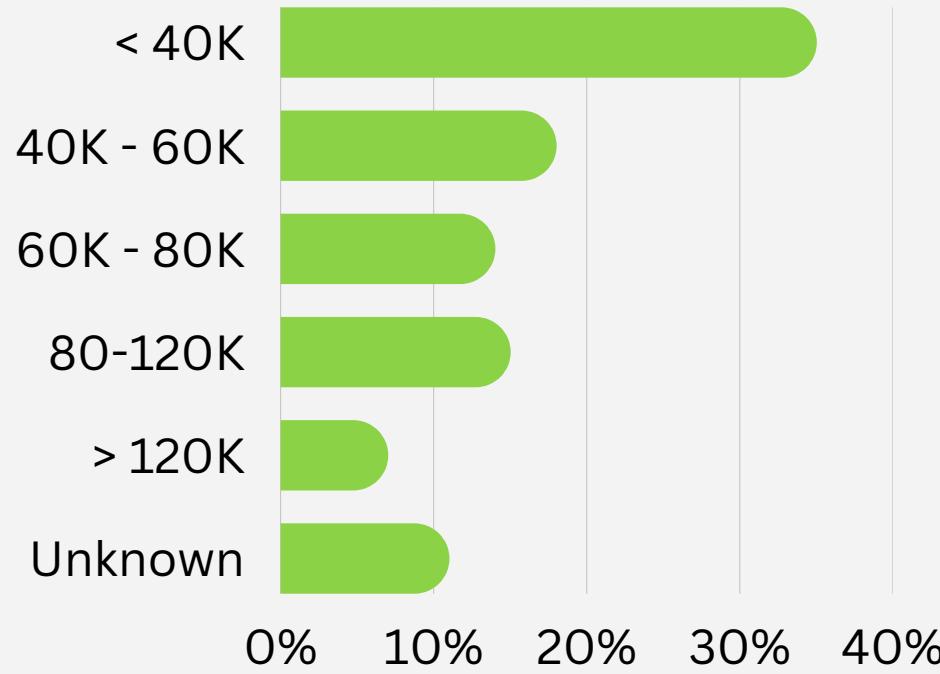
Attrited Customer



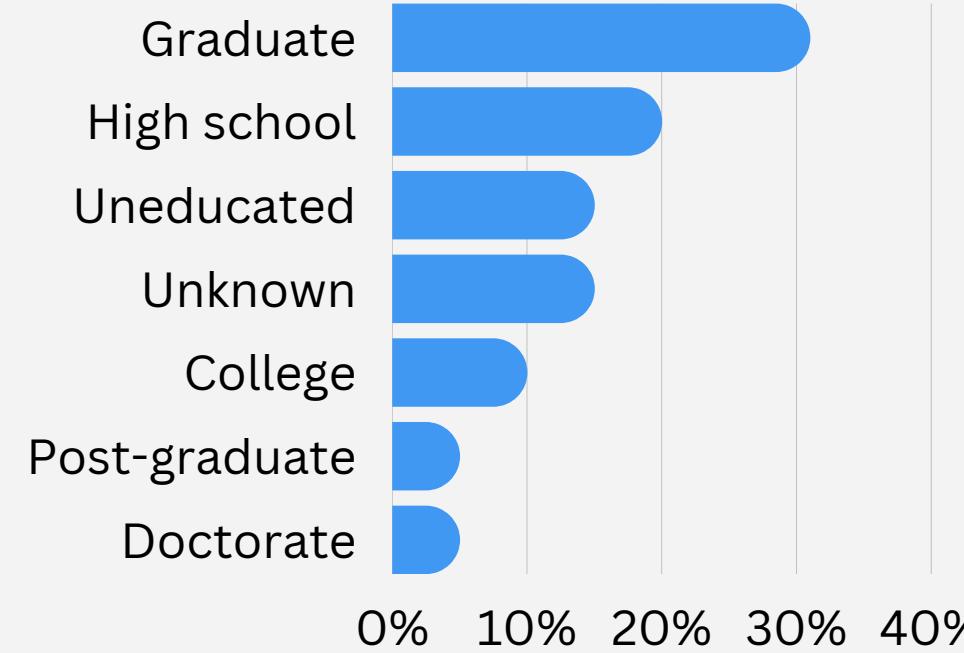
Gender & Marital Status



Income Category



Education Level



Credit Card Type





Literature Review



Previous research

- "A comparison of machine learning techniques for customer churn prediction,"
by Vafeiadis et al. (2015)



Churn prediction on a telecommunication Dataset

The authors evaluated the performance of various model:

- Logistic Regression
- Decision Trees
- Random Forests
- Support Vector Machines
- Neural Networks
- Gradient Boosting Machines



Best performance : Gradient Boosting Machines



Worst performance : Neural networks





Model

Baseline

- Majority Classification - predicts the majority class in the set
 - Random Classification - randomly assigns labels to test samples.
-

Linear Classifier

Assumes Indep. Variables normally distributed - Gaussian Naive Bayes

Binary classification - Logistic Regression 



Model

Advanced Algorithms

- Decision Tree - recursively partitioning the input space into subsets
 - Random Forest - combines bagging with feature randomness
 - Multiple decision trees
 - K-Neighbors classifier - based on the k-nearest neighbors
 - Bagging classifier - bootstrap aggregating
 - Boosting classifiers
 - Gradient boosting
 - XGBoost
 - Ada-Boost
- 



Model

S.V.C. & Neural Networks

- Support Vector Classifier - finds the best decision boundary that separates the input data into different classes

Features

- Non-linear
- Work well for high dimensional data
- Computationally expensive

- MLP classifier- an artificial neural network that consists of multiple layers of interconnected nodes.

Features

- Can capture complexity of data
 - Computationally expensive
- 

Performance Metrics



Potential performance metrics

- Accuracy = $(TP+TN) / (TP+TN+FP+FN)$
- Recall (Sensitivity) = $TP / (TP+FN)$
- Specificity = $TN / (TN+FP)$
- Precision = $TP / (TP+FP)$
- F-Score = $2 \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

Objective: Maximize correct predictions of customer who will churn

		A C T U A L	
		Positive (Churn = 1)	Negative (Churn = 0)
P R E D I C T E D	Positive (Churn = 1)	True Positive	False Positive
	Negative (Churn = 0)	False Negative	True Negative

the cost of FN could be significant as failing to predict a churn event may result in the loss of a customer



Performance Metrics



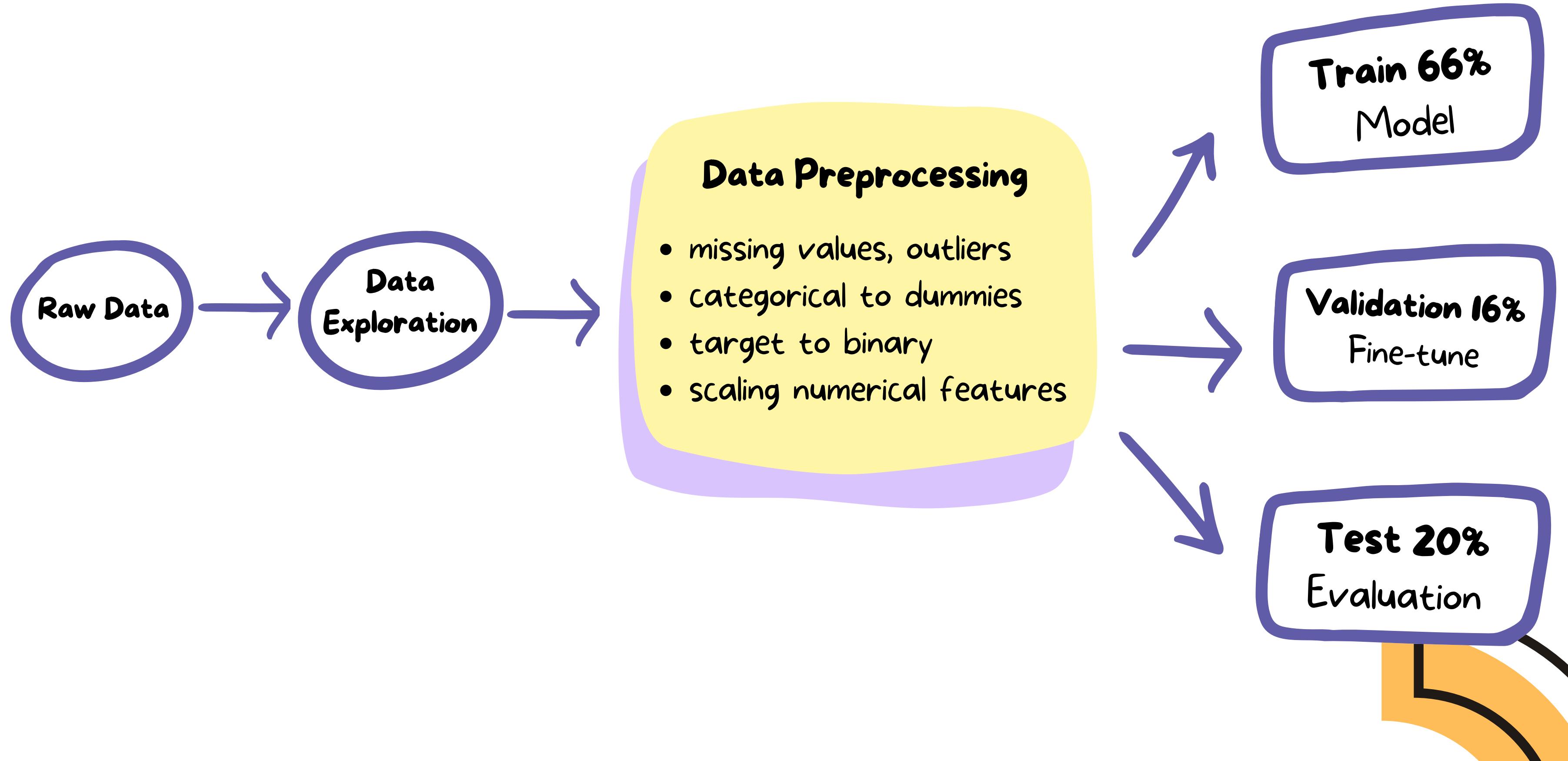
A C T U A L

	Positive (Churn = 1)	Negative (Churn = 0)
--	-------------------------	-------------------------

PREDICTED	Positive (Churn = 1)	True Positive
-----------	-------------------------	---------------

PREDICTED	Negative (Churn = 0)	False Negative
-----------	-------------------------	----------------

Data Preprocessing





Hyperparameter Tuning

- Grid-search Method -
 - Specify range of values
 - Search over all possible combinations
 - Performance Comparision -
 - Evaluate on validation sets
 - Best performance hyperparameter selected
 - Alogorithms
 - GridSearchCV
 - RandomizedSearchCV
- 

Results of models

#	classifier	Recall	Accuracy	Precision	F1	CV_score
1	Decision Tree	91.13%	89.59%	62.08%	73.85%	93.62%
2	Gradient Boosting	89.91%	97.04%	91.59%	90.74%	99.10%
3	XGBoost	88.07%	96.54%	90.28%	89.16%	99.29%
4	Random Forest	85.63%	95.36%	85.63%	85.63%	98.54%
5	Bagging	85.02%	96.25%	91.15%	87.97%	97.94%
6	AdaBoost	82.57%	95.36%	87.95%	85.17%	98.68%
7	MLP	74.01%	91.46%	73.33%	73.67%	94.13%
8	Support Vector Classification	69.72%	91.91%	77.82%	73.55%	94.47%
9	Logistic Regression	55.05%	89.93%	75.95%	63.83%	92.38%
10	Gaussian Naive Bayes	53.82%	86.77%	60.07%	56.77%	85.63%
11	K-Neighbors	31.80%	87.12%	73.24%	44.35%	80.82%
12	Baseline_Random	16.62%	51.33%	50.15%	24.96%	NA
13	Baseline_Majority	0.00%	83.86%	0.00%	0.00%	NA

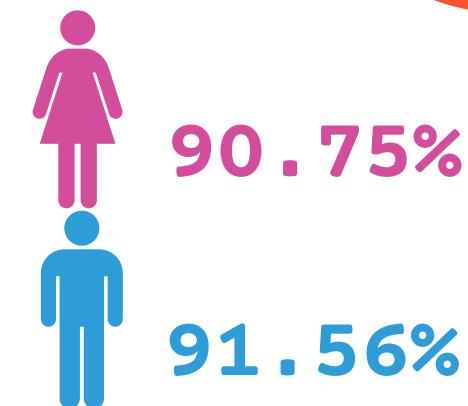
Results on another dataset

Classifie	recall (test)	f1-score (test)	cv_score (train)	accuracy (test)	precision (test)
Decision Tree	77.1%	74.4%	84.9%	92.4%	71.8%
Gradient Boosting	75.0%	80.9%	90.5%	94.9%	87.8%
Bagging	72.9%	71.1%	84.0%	91.5%	69.3%
Random Forest	70.8%	74.7%	89.5%	93.1%	79.1%
XGBoost	70.8%	79.1%	89.7%	94.6%	89.5%
Gaussian Naive Bayes	67.7%	31.6%	56.6%	57.9%	20.6%
MLP	53.1%	54.3%	79.7%	87.1%	55.4%
Support Vector Classification	46.9%	52.0%	82.3%	87.6%	58.4%
AdaBoost	41.7%	50.6%	77.9%	88.3%	64.5%
Logistic Regression	27.1%	36.4%	78.4%	86.4%	55.3%
Baseline_Random	12.8%	20.0%		48.3%	44.8%
K-Neighbors	6.3%	11.3%	63.5%	85.9%	60.0%
Baseline_Majority	0.0%	0.0%		85.6%	0.0%

Fairness

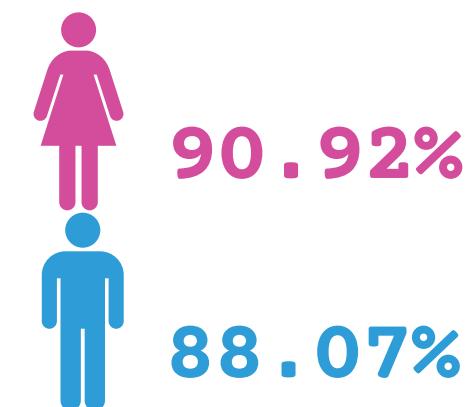
Evaluation of Final Model

Recall by Gender
Diff : 0.81%



Equal opportunity difference (EOD) score of 0.0356
There's a difference of 3.56% in TPR between two gender group

Accuracy by Gender
Diff: 2.85%

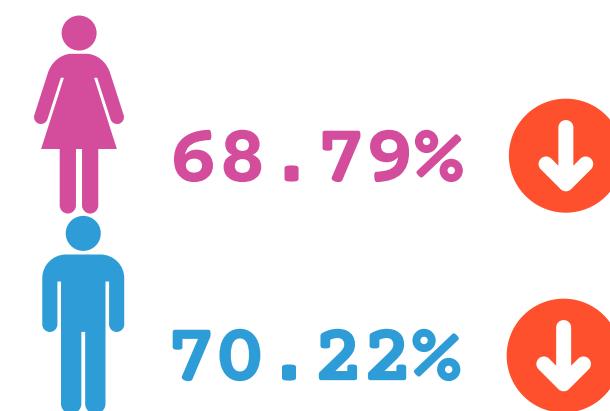


Evaluation of Final Model + Fairness Algorithm (Post-processing approach)

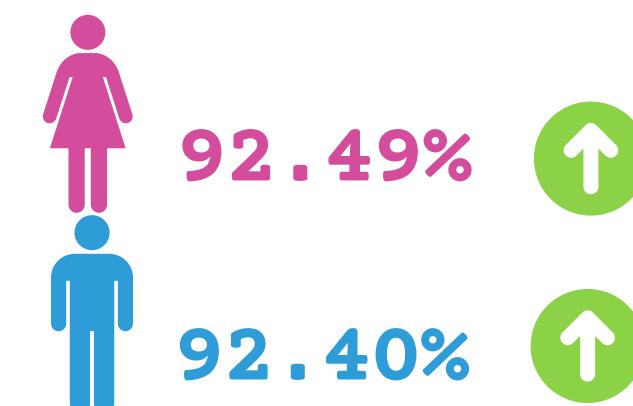
Demographic Parity (ensure that the % of positive outcomes is the same accross groups)

- Suggested threshold by algorithm = 0.80

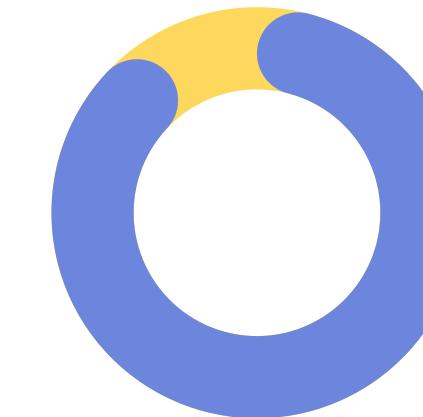
Recall by Gender
Diff: 1.43%



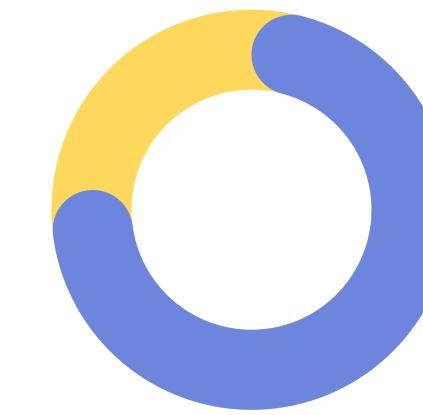
Accuracy by Gender
Diff: 0.09%



Conclusion Future Direction



Dataset 1 : Decision Tree
91.1% Recall
89.9% Accuracy



Dataset 2 : Decision Tree
77.1% Recall
92.4% Accuracy

Business context :

- interpretability, implementation
- Future Direction: fairness marketing

Thank you

