Political Polarization using NLP

TEAM LARAN
GitHub: *https://github.com/MAdnanM94/LARAN*

Problem Statement

Analyze written transcription of presidential and vice presidential debates in order to:

- Determine parlance specific to U.S. Democratic and Republican parties.
- Use this determination to classify words and phrases according to party affinity as per the predicted probability

$P(D|text) = P(Democratic|text)$

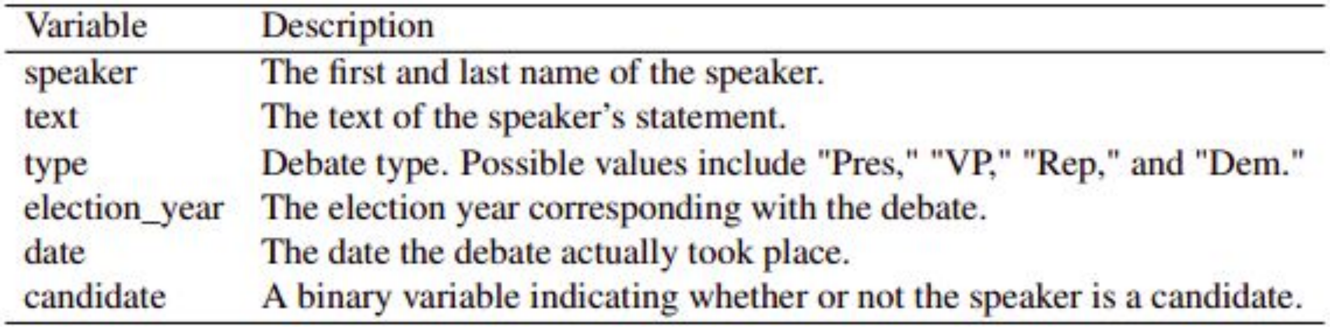
$P(R|text) = P(Republican|text)$

Such that, $P(R|text) = 1 - P(D|text)$

Description of the dataset

Dataset scraped by a third party (cited below) from website of commission of presidential debates and rev.com for Presidential, VP, primary debate transcripts since 1964.

List of columns in the dataframe:

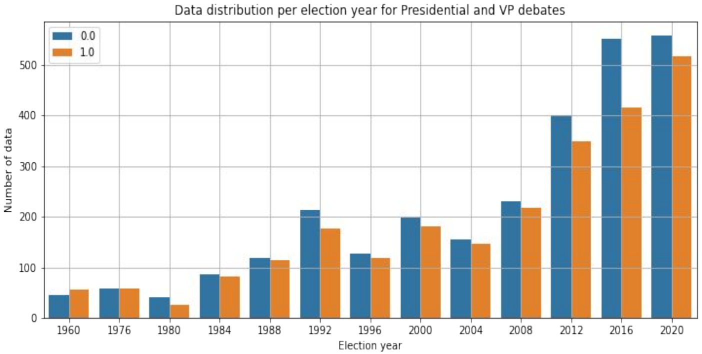


Variable	Description
speaker	The first and last name of the speaker.
text	The text of the speaker's statement.
type	Debate type. Possible values include "Pres," "VP," "Rep," and "Dem."
election_year	The election year corresponding with the debate.
date	The date the debate actually took place.
candidate	A binary variable indicating whether or not the speaker is a candidate.

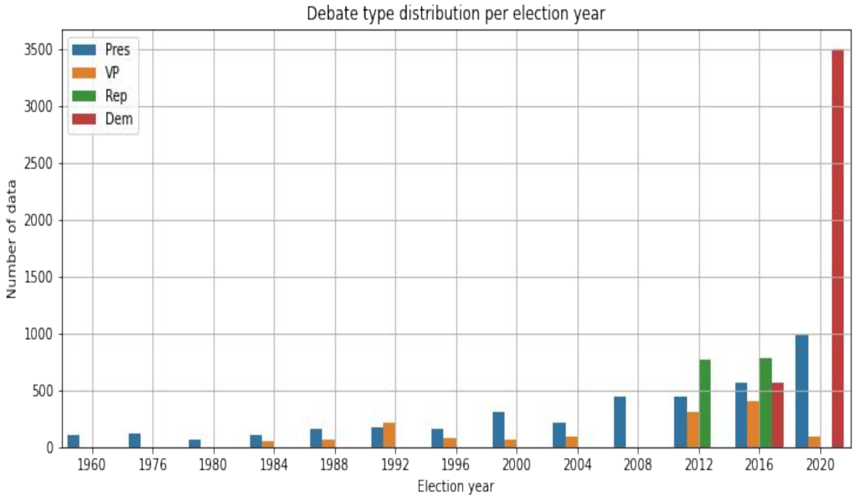
Boolean Target column added: 0 for Republican, 1 for Democrat

Source: **Martherus, James, Introducing the Transcripts of US Presidential Debates Data Set (May 27, 2020). Available at SSRN: https://ssrn.com/abstract=3611815**

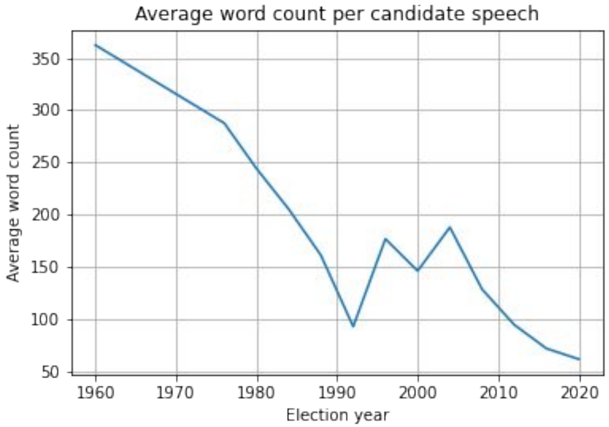
Exploratory data analysis



Data distribution per election year for Presidential and VP debates



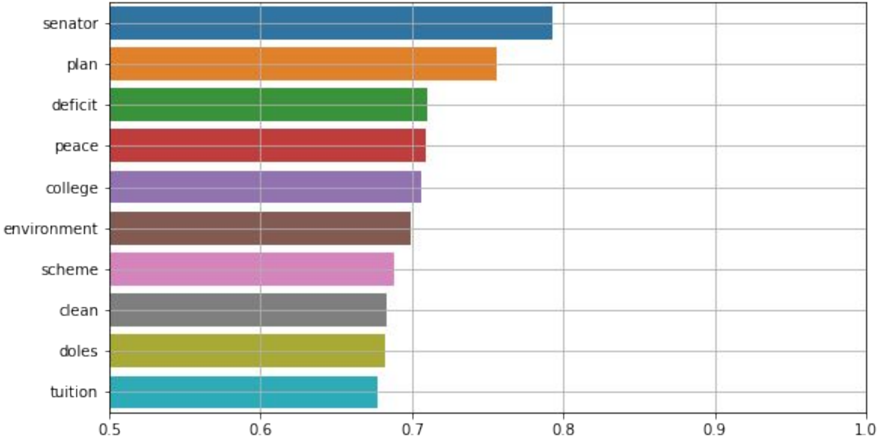
Debate type distribution per election year



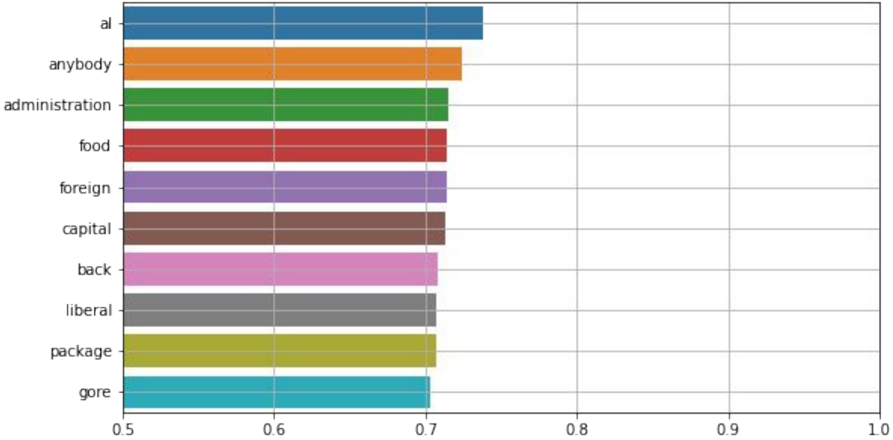
Average word count per candidate speech

Words Most Attributed to Democrat and Republican

Democrats in 1996



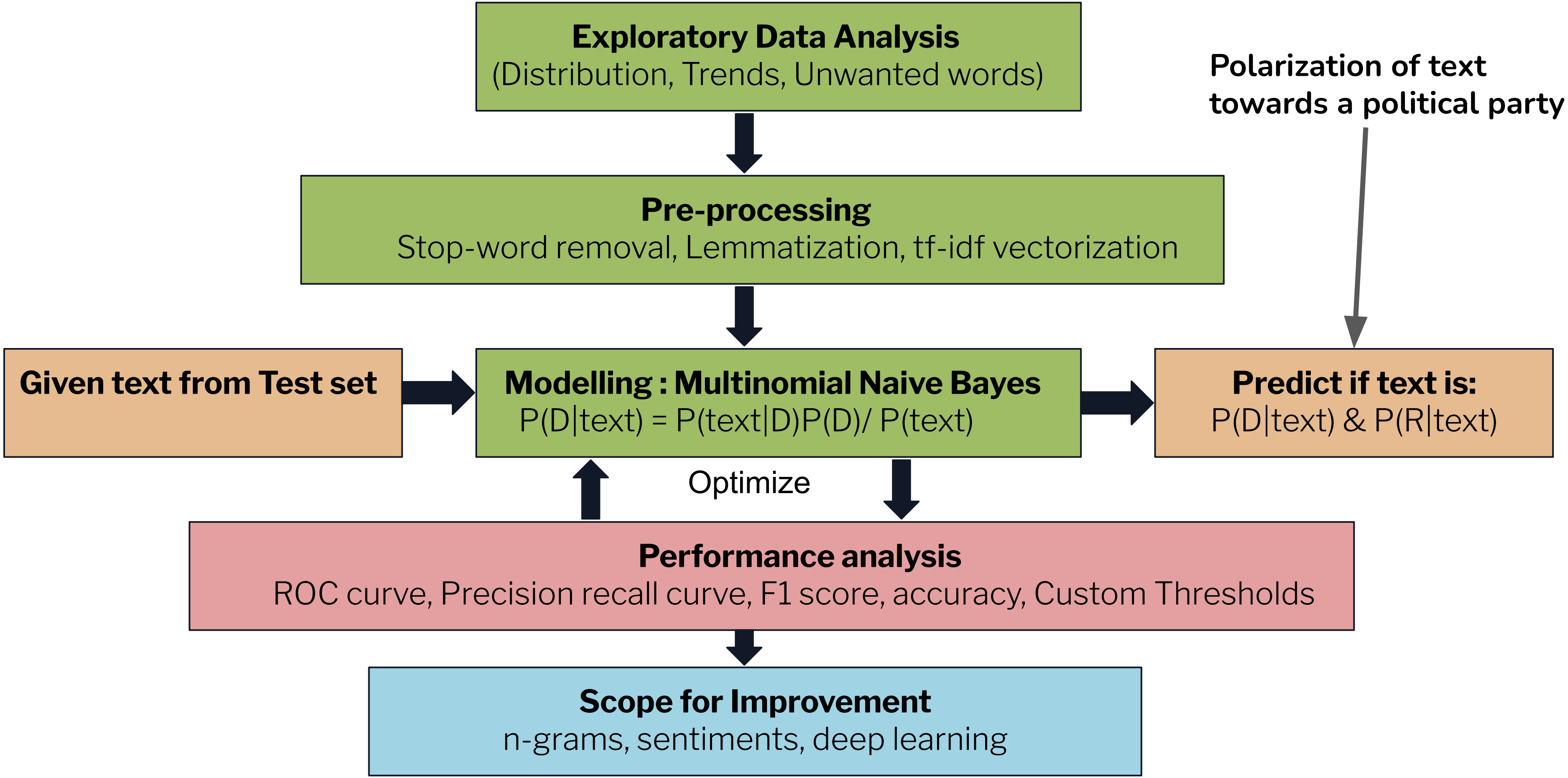
Republicans in 1996



peace, college, environment, tuition

administration, food, foreign, capital, liberal

Our Approach



Exploratory Data Analysis
(Distribution, Trends, Unwanted words)

Pre-processing
Stop-word removal, Lemmatization, tf-idf vectorization

Given text from Test set

Modelling : Multinomial Naive Bayes
$P(D|text) = P(text|D)P(D)/ P(text)$

Polarization of text towards a political party

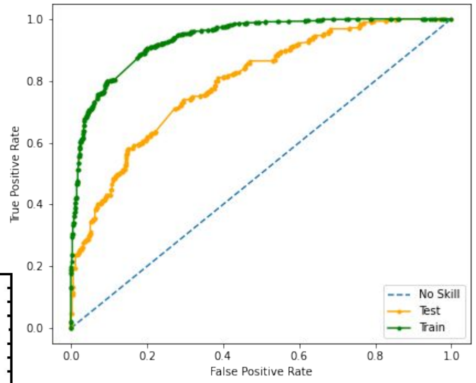
Predict if text is:
$P(D|text)$ & $P(R|text)$

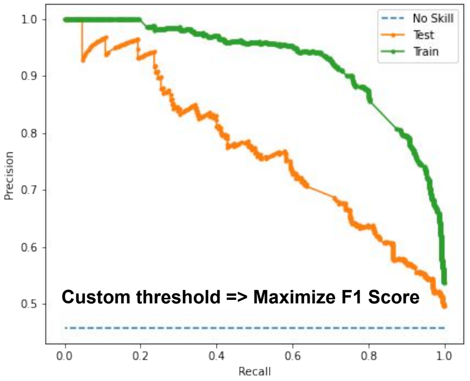
Optimize

Performance analysis
ROC curve, Precision recall curve, F1 score, accuracy, Custom Thresholds

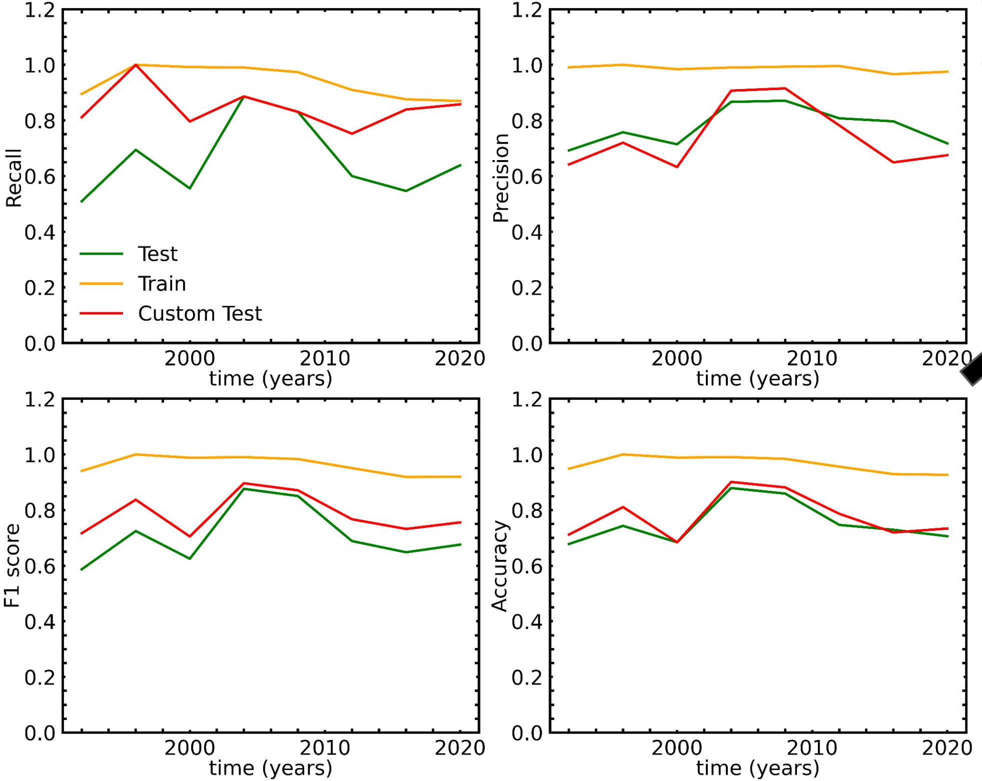
Scope for Improvement
n-grams, sentiments, deep learning

Metrics





Custom threshold => Maximize F1 Score



- We train MNB for each debate independently from 1992 - 2020

- Use our custom texts to calculate the probability that the text belongs to Democratic party

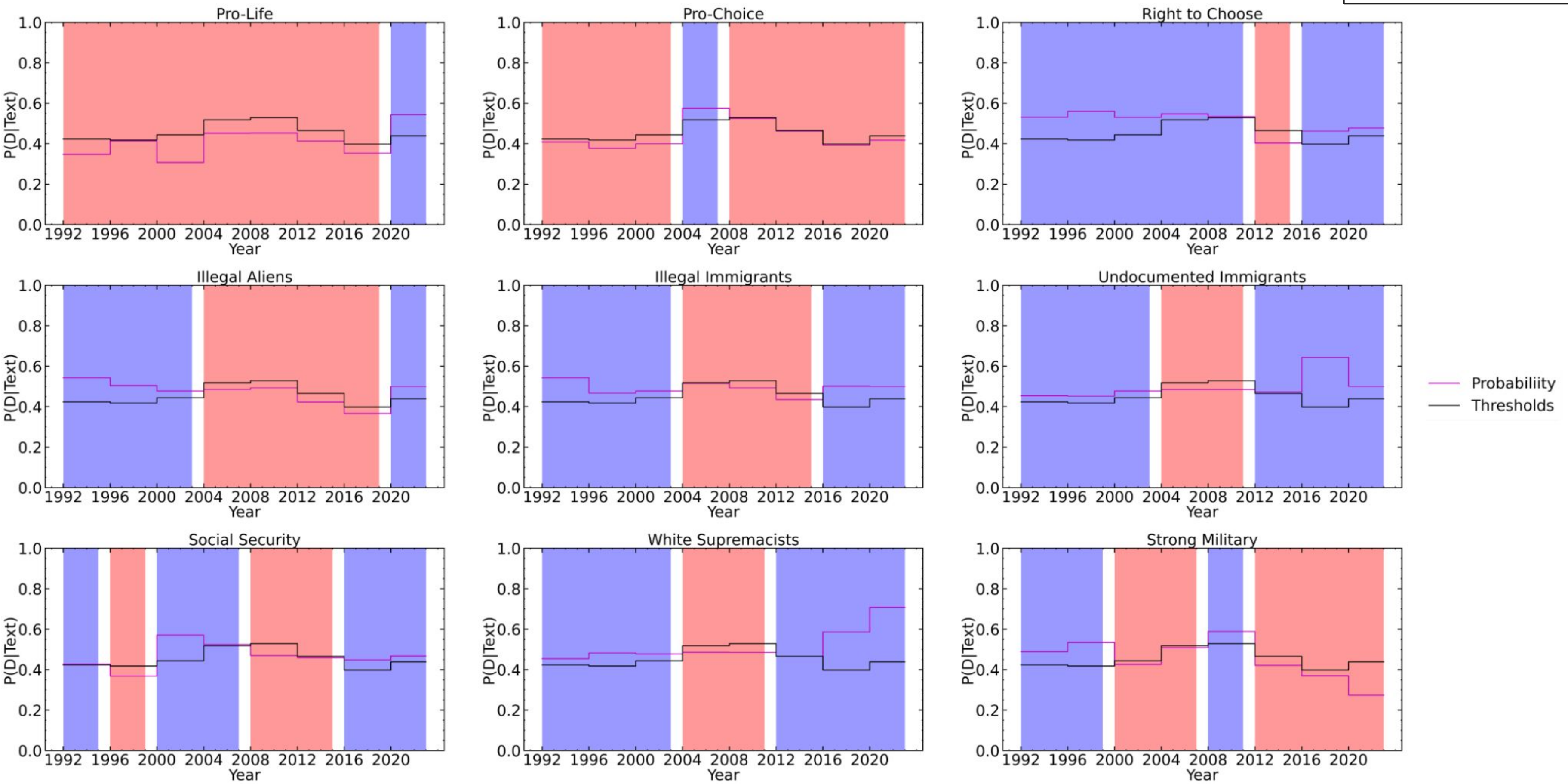
- This is given by
  P(D|text) = 1 - P(R|text)

- P(R|text) = Probability of word belonging to Republican Party

- Eventually we get our custom texts politically polarized

Polarization Across Administrations

```
IF Probability > Thresholds
   Text => Democratic (Blue)
ELSE
   Text => Republican (Red)
```



Summary and Conclusions

Build a classification system to classify text as Democrat or Republican leaning

Analyze political polarization of words across different political regimes

Key uses:

1) Politicians for constructing and promoting campaign platforms
2) Independent lobbyists targeting proposals to either party

Future Directions

Need to account for sentence structure and nuance in the debate transcripts.

- MNB assumes words are conditionally independent
- Analysis does not incorporate tone or meaning

MODEL & FEATURE ENHANCEMENT

- Look at word dependencies and track how words and phrases are used together (e.g. longer n-grams or Deep Learning)
- Incorporate feature reduction through SVM and Regularization
- Filter out words that are more neutral and common between parties

SENTIMENT ANALYSIS

- Compute sentiment polarity for debates
- See how sentiments correlate with political polarization
- Track co-movement of sentiments with approval ratings

Team Members

Lisa Berger – *Polarization analysis*

Aniket Joshi – *Data gathering, Cleaning and Processing*

Reza Averly – *Classification (BERT)*

Adnan Mahmood – *Data Visualization and Classification (MNB)*

Nikhil Ajgaonkar – *Data Visualization and Classification (MNB)*

Acknowledgments: Angelo Taranto (Project Mentor)

PRE-Trained BERT (Deep Learning) MODEL

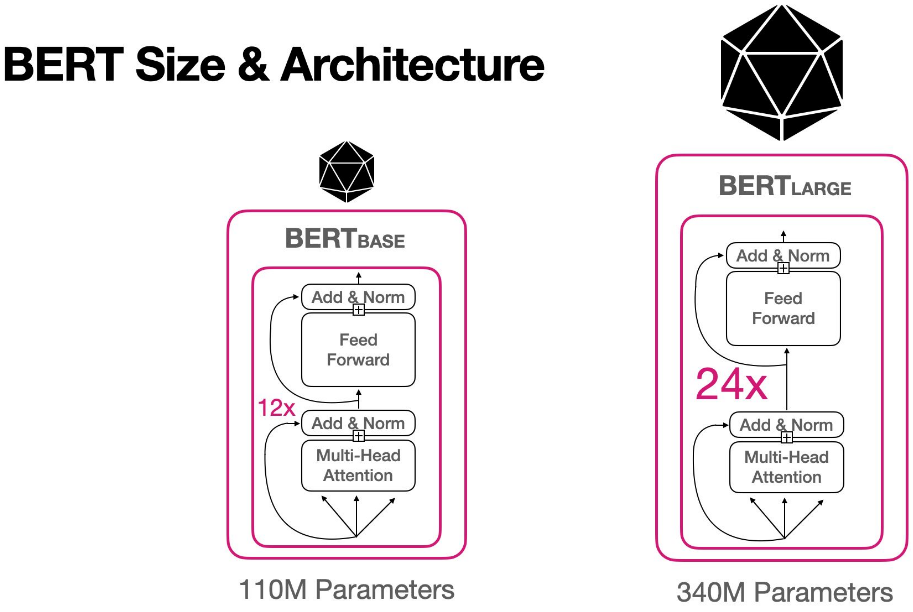
BERT (Bidirectional Encoder Representations from Transformers) is a Machine Learning (ML) model for natural language processing, developed at Google AI Language (340M parameters and 12 transformer layers). The smaller version, DistilBERT, is faster and cheaper with 110M parameters, which we used for our project.

Training hyperparameters:

Epoch	: 10
Batch size	: 16
Learning rate	: 1e-4

Performance Metrics:

Training Accuracy	: **98%**
Validation accuracy	: **82%**
Recall accuracy	: **82%**
Precision accuracy	: **82%**


BERT Size & Architecture
BERTBASE
Add & Norm
Feed Forward
12x
Add & Norm
Multi-Head Attention
110M Parameters
BERTLARGE
Add & Norm
Feed Forward
24x
Add & Norm
Multi-Head Attention
340M Parameters