# Statistical patterns in political rhetoric: A quantitative analysis of Donald Trump's 2024 election campaign speeches

## Statisztikai minták a politikai retorikában: Donald Trump 2024-es elnökválasztási kampánybeszédeinek kvantitatív elemzése

Corvinus University of Budapest

Scientific Students' Associations Conference (TDK)

Author: Barnabás Epres

Business Informatics Engineer BSc, 3rd year

Supervisor: László Kovács

Institute of Data Analytics and Information Systems

Table of Contents

# I. Introduction

Donald Trump has secured his position as the president of the United States of America for the second time in November 2024, with a shocking increase in votes compared to the previous two elections. This study aims to analyze how his election rally speeches influenced his popularity, uncovering hidden patterns, for instance whether more positive speeches had a greater impact, or whether people reacted to simpler or more complex speeches, and understanding how modern voters resonate with other specific aspects of old-fashioned lengthy live political speeches.

With conducting this research, I also offer a seldom discussed point of view into the emerging trend of studies about the effect of social media on public political opinion forming, with asking the question whether campaign speeches are still directly correlated with politicians' popularity. As Ntontis et al. (2013) argues, after the lost election of 2020, Donald Trump has turned to an even more populist leaning rhetoric which eventually led to the attack on the Capitolium (Ntontis, et al., 2023). I was curious whether the same rhetorical elements persisted in the 2024 election campaign.

# II. Descriptive statistical analysis

## II.1. Data

Collecting and preprocessing data is a crucial part of any statistically heavy research analysis. The results depend highly on the quantity, and especially on the quality of the data. During my research I have collected data from various sources. All sources which are not referable research papers (speeches, code) are in the GitHub repository, referred in the bibliography.

### II.1.1. Data collection

In this study Donald Trump's speeches are analyzed that were given at the general election rallies in 2024 between the 6th of June and the 4th of November. Originally 80 speeches were given, however only those were collected, which were separatable by either the state it was held in, or the date it was held on. In case of two speeches on the same day in the same state or city, only the longer speech was chosen, to avoid duplications in latter statistical analysis. Moreover the 8th speech held on the 13th of July in Butler, Pennsylvania has been ignored due to assassination attempt against the candidate. In total, 73 speeches have been selected as part of the research. Each speech lasted for around 1.5 - 2 hours, making it a total of approximately 130 hours.

On the technical side, the rallies were only available in video format uploaded by multiple streaming channels to YouTube. To receive useful text data from the videos I have built an extractor pipeline using Python. The first step of my technical solution was converting the mp4 format videos to mp3 sound files. After having converted the data, I utilized the OpenAI developed Whisper model (exact model: openai/whisper-large-v3) to create the transcription of the sound files. Whisper is widely regarded as one of the most effective open-source multi-language transcriber and text-to-speech model available (Graham & Roll, 2024).

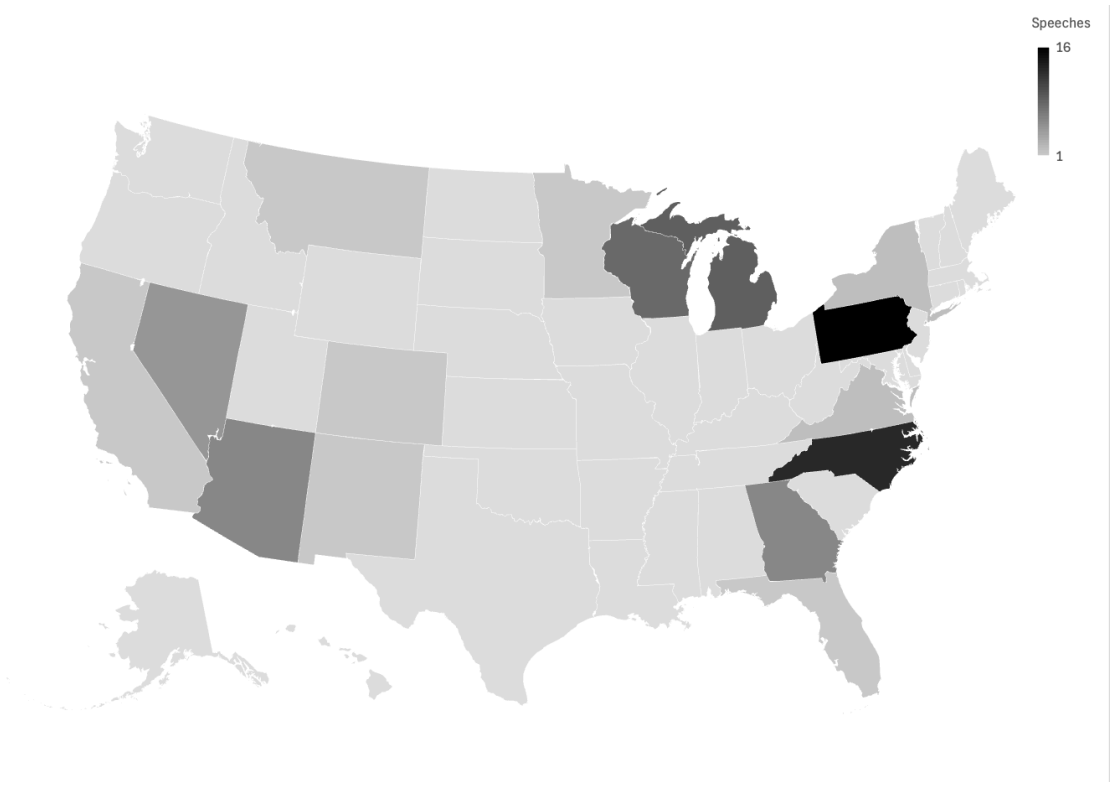


*Figure 1: The distribution of speeches among the states of the USA. Created with Bing AI*

As the map depicts, most speeches were given in swing states. The four states with the largest number of speeches were Pennsylvania (16), North Carolina (13), Michigan (9) and Wisconsin (8). The frequent appearances in the above-mentioned states turned out to be successful, because Donald Trump has won in all four mentioned states in the 2024 presidential elections.

### II.1.2. Data preprocessing

To construct the converted speeches - also known as the corpus - into correct format for text mining, a couple of preprocessing steps were needed to be taken. The following changes were computed using R programming language (version: 4.4.2), taking advantage of its widespread solutions for data wrangling (Boehmke & Greenwell, 2020).

As a first step, I have tokenized the text data, meaning that all sentences and words were split up until reaching the smallest interpretable word or word element. Removing punctuation marks, numbers, symbols and converting words to lowercase is also part of the technically accurate tokenization. The next steps for preparing the corpus for the document-term-matrix includes

removing stop words and word stemming. Stop words are words (or tokens in my case) used in natural speech to enhance the flow of the text but have no contribution when it comes to text mining, for instance words like “the” or “with”. On the other hand, stemming focuses on detaching the root word from possible suffixes, so that for example ‘apples’ and ‘apple’ are not counted as two separate tokens (Silge & Robinson, 2024).

The prepared data was converted to a document-term-matrix, where each row represents one text from the corpus – one speech in this case – and each column is a separate token. The values are the number of appearances of each token in each text, making the matrix ready for further statistical analysis. In *Table 1*, the first rows of the document-term-matrix are present, showing that in the first speech the word *thank* and *much* has appeared 75 and 38 times respectively, while the word *phoenix* has appeared 4 times, suggesting that the first speech has taken place in the city of Phoenix, the capital of the state of Arizona.

| *docid* | *thank* | *much* | *hello* | *phoenix* | *...* |
|---|---|---|---|---|---|
| *text1* | 75 | 38 | 4 | 4 | … |
| *text2* | 11 | 11 | 0 | 0 | … |
| *text3* | 36 | 31 | 1 | 0 | … |
| *text4* | 39 | 27 | 3 | 0 | … |
| *...* | … | … | … | … | … |

*Table 1: The first 5 columns and rows of the document-term-matrix*

## II.2. Exploratory analysis

Most common words in this case simply mean the words which appeared the most in the document-term-matrix. As per section II.1.2., the corpus has been tokenized, stemmed and stop words has been removed, therefore it is possible, that merely the common words could be a good baseline for getting to know the speeches, because words which are irrelevant when it comes to text-mining - such as “the”, or “a” - are not accounted for.

| *Words* | *Appearance* |
|---|---|
| go | 10539 |
| know | 7388 |
| peopl | 5426 |
| countri | 4820 |

| | |
|---|---|
| said | 4748 |
| want | 4460 |
| like | 4163 |
| get | 3909 |
| great | 3733 |
| one | 3451 |
| say | 3437 |
| think | 3230 |
| right | 3181 |
| thank | 3021 |
| got | 2780 |
| presid | 2771 |
| just | 2749 |
| now | 2608 |
| year | 2493 |
| thing | 2477 |

*Table 2: The 10 words with the most appearance in the document-term-matrix in descending order*

However, naturally, in such large document-term-matrix the most common words become generalized easily, and frequent natural speech words dominate the list of most common words. The words “peopl”, “countri” and “presid” indicate that the broad topic of the corpus is possibly related to politics, nevertheless one could not dig any further using only this list.

In the previous example, I have listed the most common words ordered by the number of appearances in the document-term-matrix, in other words, I have calculated the term frequency of all terms present in the matrix. To counterweight the effect of generalized term frequencies and receive the unique, yet important words in a matrix, the number of different documents which they appear in should also be accounted for. However, the less document a token appears in, the more outstanding that token is, therefore while calculating the combination of term frequency and document frequency, the latter shall be inverted in the equation. Therefore, general words like “thing” will receive lower weights, while words like “Wisconsin”, which only appears in speeches

given in the state of Wisconsin will receive higher weights. The equation for calculating the weights of term frequency inverse document frequency (TF-IDF) weighted document-term-matrix is the following, where *tf* is the term frequency, *N* is the number of documents in the matrix and *df* is the document frequency (Silge & Robinson, 2025).

$$w_{i,j} = tf_{i,j} * \log\left(\frac{N}{df_i}\right)$$

The following table displays the 10 words with the highest TF-IDF score across the document-term-matrix, alongside with the document frequency.

| *Words* | *TF-IDF score* | *Rank* | *document frequency* |
|---|---|---|---|
| glori | 231.416122 | 1 | 8 |
| raimondo | 152.792475 | 2 | 1 |
| arnold | 80.122883 | 3 | 1 |
| wisconsin | 76.1581637 | 4 | 28 |
| michigan | 70.902538 | 5 | 41 |
| plant | 70.5970764 | 6 | 45 |
| nevada | 65.083754 | 7 | 12 |
| secretari | 64.5623297 | 8 | 26 |
| mike | 63.2333983 | 9 | 31 |
| doctor | 62.6802543 | 10 | 32 |
| chart | 62.6621772 | 11 | 38 |
| bobbi | 58.4565058 | 12 | 24 |
| snake | 57.0532899 | 13 | 5 |
| tommi | 57.0205899 | 14 | 7 |
| kamala | 56.3895238 | 15 | 68 |
| starlink | 54.9495104 | 16 | 19 |
| commonwealth | 54.3649411 | 17 | 19 |
| indict | 54.1736186 | 18 | 22 |

| | | | |
|---|---|---|---|
| garbag | 52.2923305 | 19 | 18 |
| comrad | 51.4178244 | 20 | 16 |

*Table 3: The 10 words of the document-term-matrix with the highest TF-IDF score*

Evidently, most, if not all words are immediately recognizable elements of Donald Trump's campaign speeches. The list features Wisconsin, Michigan and Nevada, all three states were important locations during the campaign rally. Moreover, important personnel of the rally also arise in the list, "raimondo" refers to Gina Raimondo, commerce secretary of the USA, "tommi" refers to Tommy Thompson, former republican Wisconsin governor and "kamala" refers to Kamala Harris, his opponent candidate on the election campaign.

## II.3. Detailed quantitative analysis – methodology

Detailed quantitative analysis is needed for further analyzing the speeches. Advanced methods reveal underlying information of the campaign speeches, while approaching the speeches purely from a statistical point of view. The aim is to gain understanding of certain characteristics of the speeches, for instance whether they were more positive or negative leaning, were they lexically diverse or how similar were the speeches to each other.

### II.3.1. Lexical diversity

The concept of lexical diversity focuses merely on the ratio of unique tokens, also known as types, to the total number of tokens in the corpus. The most straight forward approach to receive the lexical diversity of a corpus is to calculate the following equation:

$$\mathrm{TTR} = \frac{\text{Number of types}}{\text{Number of tokens}}$$

However, this method fails when it comes to comparing two corpuses of different length. In practice, the tokens tend to be repetitive, therefore the number of types reaches an upper threshold after a certain corpus length, while the number of tokens is only limited by the mere length of the corpus. Thus, the longer a corpus is, the faster its TTR score converges to zero. To calculate comparable lexical diversity scores, the equation needs to be corrected.

$$\mathrm{CTTR} = \frac{\text{Number of types}}{\sqrt{2 * \text{Number of tokens}}}$$

The corrected type to token ration (CTTR) calculates the lexical diversity in a comparable way. In theory, CTTR is still, to some degree, dependent on the length of the corpus. However, in this case I consider this method to be accurate for comparison, due to having collected speeches of approximately the same length. The theoretical minimum value for the corrected type to token ratio is very close to zero, assuming a lengthy text with incredibly limited vocabulary. On the other hand, however, there is no theoretical maximum value, the value increases with the number of unique tokens (Torruella & Capsada, 2013).

### II.3.2. Jaccard-similarity

Jaccard-similarity, also known as Jaccard-index is a widely used method for calculating the similarity of two objects. In natural language processing (NLP) Jaccard-similarity is used for both search engine optimization and text mining. The simple idea behind this method is that two texts are more similar if they have more words, or tokens in common. As the equation indicates, the score is a value between 0 and 1, with 0 meaning that the two documents had no tokens in common, while 1 meaning that the two documents consist of the same tokens.

$$J_{(doc1,doc2)} = \frac{doc1 \cap doc2}{doc\ 1 \cup doc2}$$

Other similarity comparison methods exist, the most popular one is cosine-similarity which encompasses the vectorized form of two texts and calculates the angle of which the space made of the dense *n* dimensional vectors of the two text embeddings forms, and the smaller angle they form the closer they are semantically. However, this method is generally less accurate for comparing large documents, therefore I have chosen to use Jaccard-similarity, which is not length-dependent (Zahrotun, 2016).

### II.3.3. Political scaling

In political text mining, document scaling is one of the most interesting and informative method of extracting information from texts. Scaling, or positional modelling of documents means positioning them on given ranges, for example libertarian – authoritarian, or political left – right.

During my research I have attempted to position Donald Trump's campaign speeches on the widely used democrat – republican line, which in my opinion fits the best, when it comes to American political scenery. The traditional European left, right, socialist or even green ideologies

are not present in the political environment of the United States as they are in Europe, however there is a strong borderline between democrat and republican ideologies. Due to the fact the Donald Trump was the candidate of the republican party, I assume, that Donald Trump's speeches will most likely appear republican leaning.

One of the most common statistical methods for scaling is wordscores. As Love (2008) reveals**,** the idea behind this method, is that similar political texts contain alike unique words, which could form a linear connection. Establishing the scale, two datasets consisting of texts from the two ends of the linear scale are measured, and each word receives a score based on its individual uniqueness towards one end of the scale, creating reference scores. The "virgin texts" are fitted on the word scale created by the reference texts, and each word in a text is assigned a value based on the relative distance from the two ends of the scale (Lowe, 2008).

## II.4. Detailed quantitative analysis - results

### II.4.1. Lexical diversity

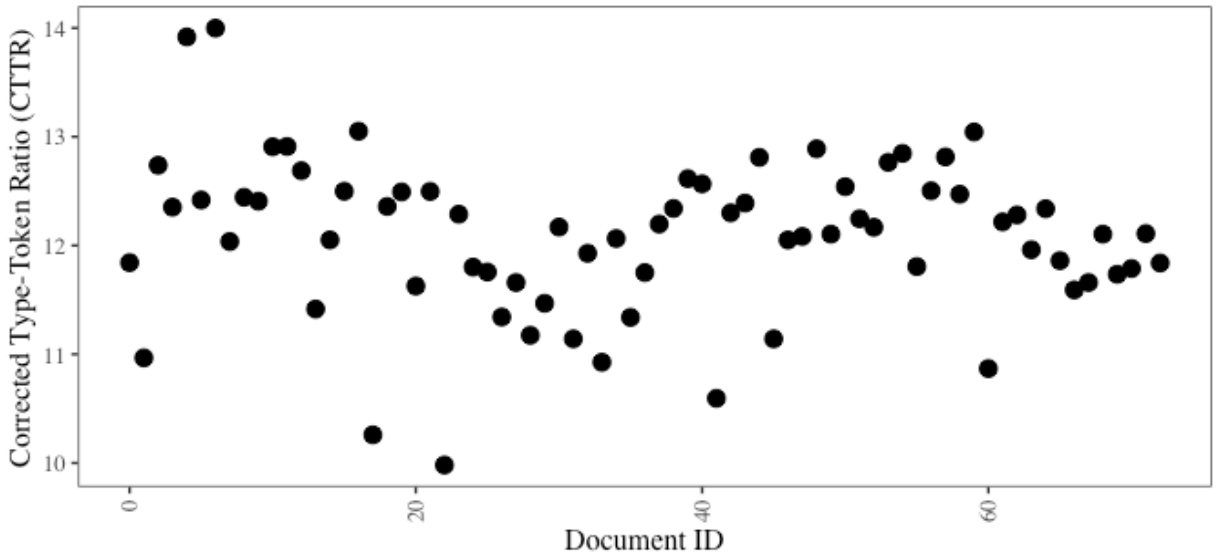


*Figure 2: The corrected type-to-token ratio of the documents in a chronological order*

The corrected lexical diversity scores vary little, based on the plot. The mean, the minimum and the maximum are 12.08, 9.98, 13.99, respectively. The corrected lexical diversity scores of

Donald Trump's campaign speeches follow no trend either based on the plot, meaning, that there is no immediately recognizable pattern between the speeches and their lexical diversity.

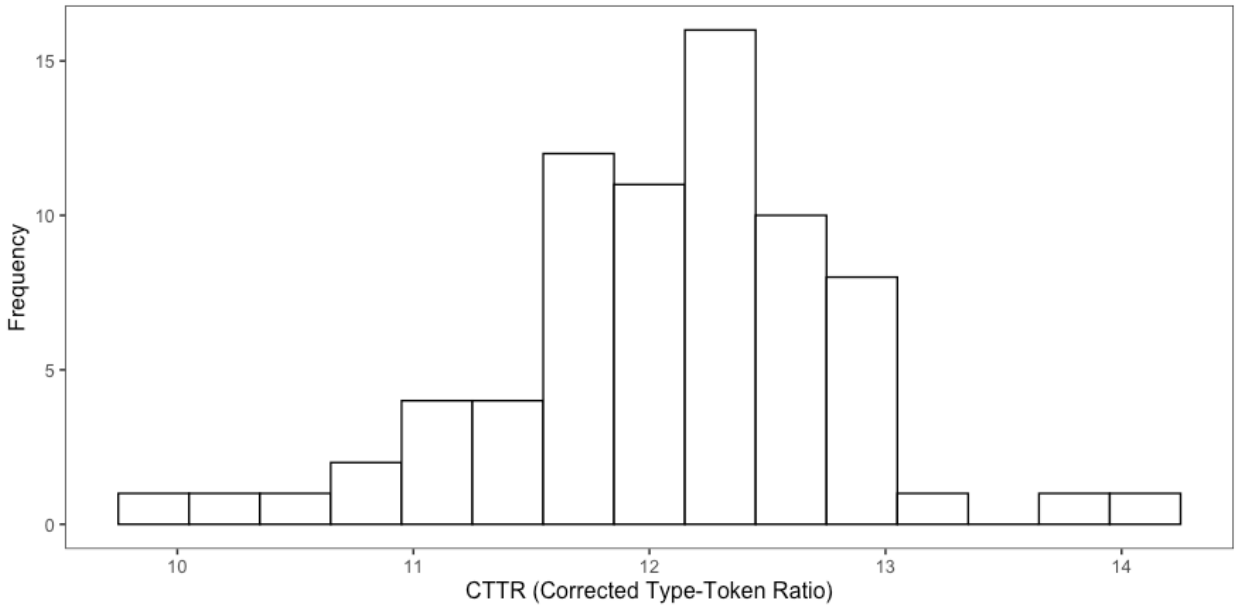


*Figure 3: The distribution of the corrected type-to-token ratio among the documents*

According to the histogram, the corrected type to token ratio scores follow a normal-like distribution, implying, that the scores vary around a fix average with fix variance. The Shapiro-Wilk test for normal distribution also determines the CTTR scores to be normally distributed on 5% of significancy. Moreover, the variance of the lexical diversity scores is 0.499, which indicates, that the calculated scores typically occur in a relatively small range. The normality and a little variance can be interpreted as tough the speeches did not differ much based on the corrected type to token ratio.

Corrected lexical diversity scores of speeches given by the same person are interpretable only in terms of change over time, or variance of the scores, it is difficult to see whether a speech is lexically diverse, based only on lexical diversity scores. To grasp the real diversity, the speeches needed to be compared to other political speeches within the same domain. The corrected lexical diversity score has been calculated for Joe Biden's 2024 speech at the Democratic National Convention in Chicago as well as for Barack Obama's speech given at Kamala Harris' election campaign rally one day before the elections. The corrected lexical diversity scores were 13.5 and

13.2 respectively, which both are larger than Donald Trump's average on all standard significancy level, based on the conducted one-sample t-test.

### II.4.2. Jaccard-similarity

My hypothesis is, that while calculating the similarity index of Donald Trump's campaign speeches, two speeches given in the same state, especially in swing states will show more similarity than two speeches given in different states. Moreover, as the small variance of lexical diversity suggests, that overall, the speeches are at least moderately alike. The Jaccard-similarity scores have been calculated for all possible combination of speech-pairs, a total of 2628 speeches. The distribution of the speech-pairs takes the following form.

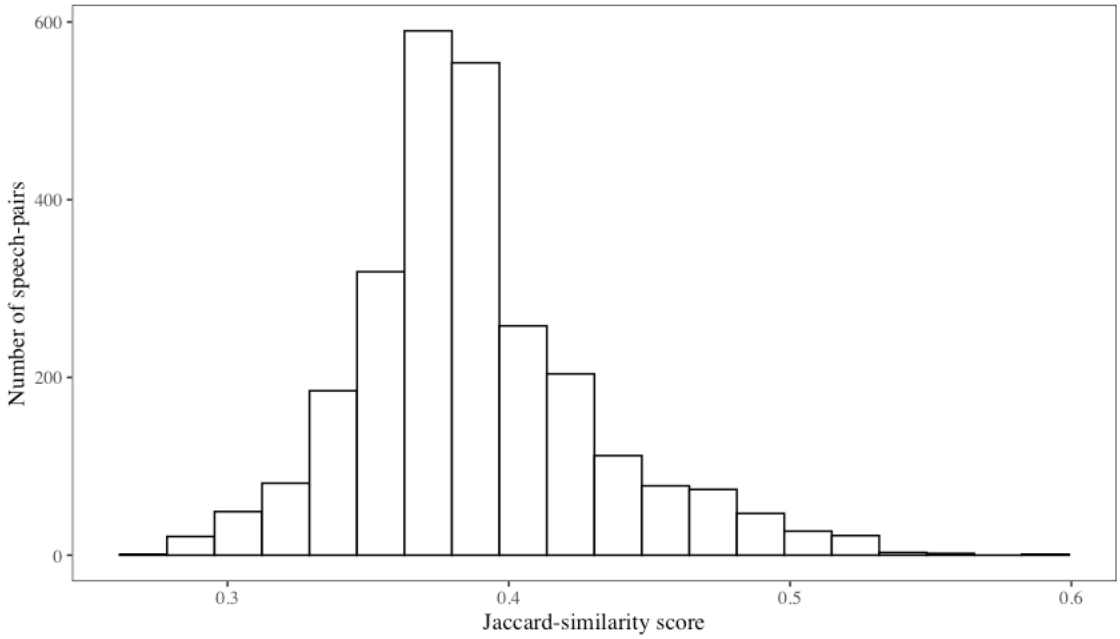


*Figure 4: The distribution of the document-pairs based on the received Jaccard-similarity index*

Results show that the average similarity of the speech-pairs is 0.387. If two speeches are randomly picked and examined, we can expect that around 38,7% of all tokens are to be found in both texts. Considering, that Jaccard-similarity was calculated using the document-term-matrix, where stop words have been removed, 38,7% of similarity could be considered moderately high (Zahrotun, 2016). Moreover, considering that all speeches had quite lengthy transcriptions, 38,7% of similarity could be considered even higher. Shorter texts are statistically more likely to receive higher score, this is because the denominator of the Jaccard-index equation is the union of the two compared texts. The longer the compared texts are, the larger their union will be and the smaller

the Jaccard-similarity will become. This conclusion also aligns with the examined lexical diversity scores, because its low variance implies less unique tokens in the documents, which generally could result more tokens in common, therefore a higher overall similarity score.

However, my initial hypothesis stated that speeches given in the same states will be similar, due to addressing the same political or socio-economic problems on multiple occasions. Empirical evidence from the Jaccard similarity scores suggests otherwise. There is no direct indicator of the above-stated hypothesis. The following table shows which state pairs had the most similar speeches on average, for instance, the speeches given in Arizona were most like the speeches given in California, second and third most alike to speeches given in Colorado and New York state, respectively. In none of the states were the speeches first, second or third most alike pair speeches given in the same state.

| *State* | *Most similar* | *Second most similar* | *Third most similar* |
|---|---|---|---|
| *Arizona* | California | Colorado | New York |
| *California* | Colorado | Nevada | Georgia |
| *Colorado* | Nevada | Virginia | Minnesota |
| *Florida* | Virginia | Michigan | Wisconsin |
| *Georgia* | New Mexico | California | New York |
| *Michigan* | New Mexico | California | Georgia |
| *Minnesota* | North Carolina | Michigan | Virginia |
| *Montana* | Minnesota | North Carolina | Virginia |
| *North Carolina* | New Mexico | California | New York |
| *New Mexico* | California | Georgia | New York |
| *Nevada* | New Mexico | California | Georgia |
| *New York* | California | Georgia | Colorado |
| *Pennsylvania* | New Mexico | California | New York |
| *Virginia* | New Mexico | New York | Georgia |
| *Wisconsin* | New Mexico | California | New York |

*Table 4: The first, second and third most similar state for each state based on the average Jaccard-similarity index of the speech-pairs*

## II.4.3. Political scaling

As I mentioned in *Section II.3.3.*, I have attempted to place the campaign speeches on a democrat – republican scale. To establish the political scale itself, for the democrat end I have chosen former president John F. Kennedy's "Peace Speech", given on the 10$^{th}$ of June, in

Washington DC. The speech is remembered as one of his most fulfilling and influential one (Sitara, 2023). On the contrary, for the republican weights in the model, I decided to utilize former president George W. Bush's 2001 inaugural speech, where he addressed the United States for the first time as president. It was a characteristic speech, depicting republican values and was highly regarded at the time (Kusnet, 2002). The presidents for the political scaling have been chosen based on a comprehensive ranking by Wicklin, (2018), where the former presidents of the United States have been ranked based on presidential greatness and political ideology (Wicklin, 2018).

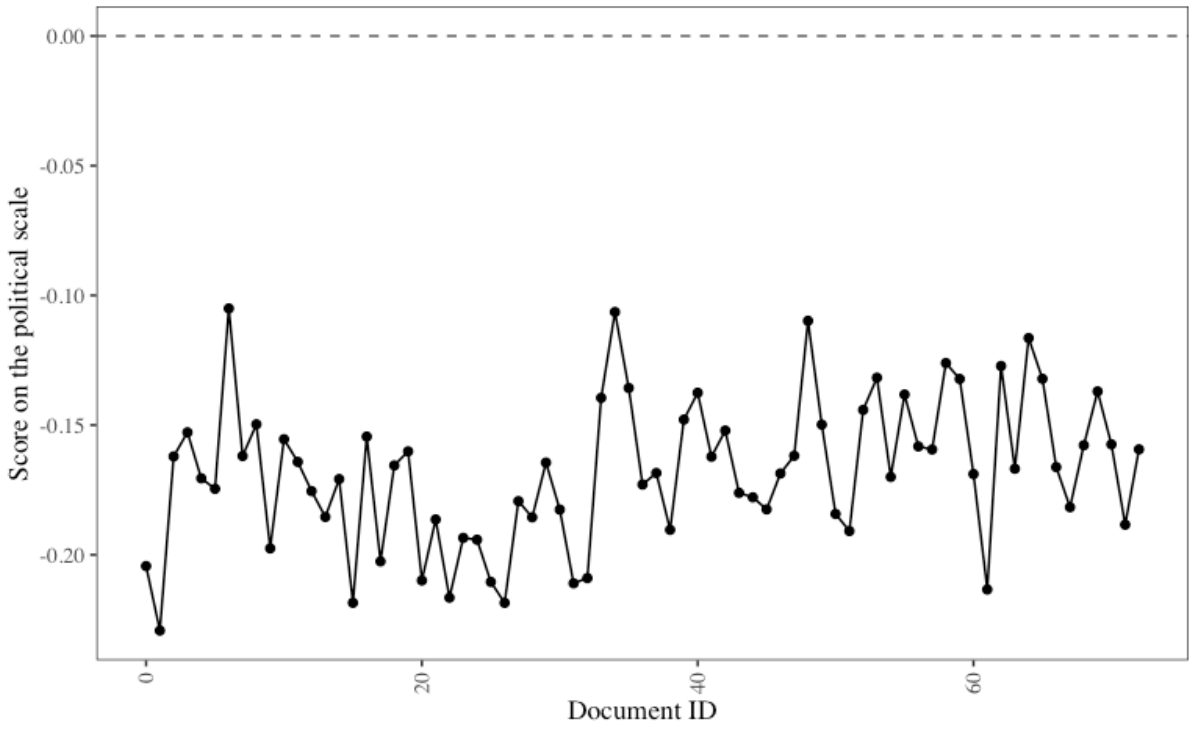


*Figure 5: The score on the republican - democrat political scale for each document in a chronological order*

The wordscores method operates with a scale of [-1, +1]. A text could mathematically reach a score of -1 or +1, if it contained solely unique words to one end of the scale. However, in practicality, a score outside of the [-0.4, +0.4] range already indicates a considerable leaning towards one end of the scale, because it means, that 40% of all unique tokens in a text (including neutral words as well) belong unquestionably to one end of a scale, which is statistically difficult reach when analyzing lengthy speeches without cropping them into smaller chunks. For reference, George W. Bush's speech received a score of -0.57, thus implying the negative range being republican. John F. Kennedy's peace speech is +0.57.

As *Figure 5* shows, the Trump campaign texts are in a range between -0.25 and -0.1, suggesting a slight or moderate republican characterizing, because for speeches of this length it is mathematically very difficult to reach a closer score to the baseline speech. Different to the previous approaches, the received score on the republican-democrat scale shows significant variance, meaning that some speeches were recognizably more republican leading, while others leaned heavily towards scale-neutrality. Interesting to point out the speeches after his assassination attempt (speeches 13-30), where the speeches tend to be heavily republican leaning. A possible explanation is that Donald Trump mentioned the attempt alongside with US gun laws frequently, which could affect the model's results. This proposes an important question, whether the neutrality of the speeches or the serious republican leaning affected more the popularity of Donald Trump.

## II.5. Political sentiment analysis

One of the most widely used and acknowledged text mining technique is sentiment analysis, which encompasses multiple approaches to classify the sentiment of texts to either positive, negative or neutral, or in some cases into different other categories (Bing, 2022).

The two main approaches to sentiment analysis are dictionary based, and machine learning based. The latter is a broader topic, it includes multiple methods such as Support Vector Machine classification, Naïve Base model approach or other deep learning algorithms. The advantage of the complexity is that classifying can be fine-tuned to a certain corpus, providing more accurate results, e.g.: at detecting sarcasm or irony. On the other hand, dictionary-based sentiment analysis is computationally less demanding, however the result is not necessarily less accurate. This method operates with pre-labeled sentiment dictionaries, which incorporate most words and an assigned sentiment. The mentioned assigned sentiment differs from dictionary to dictionary, either it may be a score of how positive or negative a word is (the range of the score may differ as well), or an emotion is assigned to the words from a pre-set list of possible emotions. Evidently, the method of calculating the sentiment score of a certain text, or part of a certain text - for example a sentence or word-pairs (bigrams) - varies based on the characteristics of the dictionary (Bing, 2022).

For analyzing the sentiment of Donald Trump's campaign speeches, I have utilized the Lexicoder Sentiment Dictionary (Young & Soroka, 2012), which is a sentiment analyzer dictionary specifically labeled for investigating political texts, and especially political speeches. The dictionary contains 2858 negative labeled and 1709 positive labeled words, moreover, a further set

of, so-called, negation of negative and negation of positive words are also included. Negation of negative words are generally positive words which are used in political context to convey a negative sentiment. On the other hand, negation of positive words is the opposite, generally negative words used to convey a positive sentiment, for example to downplay the success of the opposite candidate. The final sentiment scores for each document are the relative frequency of the sentiment in the document, for instance, the final positive sentiment score is the number of tokens labeled as positive divided by the total number of tokens in the document. Approaching the sentiment analysis problem using the Lexicoder Sentiment Dictionary might create more accurate results, because the specific politics-related words are labelled more accurately, than when using a general sentiment dictionary.

Important to note, that in case of Trump's campaign speeches the ratio of negation of negative and negation of positive words is low, almost non-existent in the corpus. A possible explanation concludes that campaign speeches lack words which are used almost exclusively by politicians, because they aim to address the public, and not to address other politicians as for instance in congress speeches. Therefore, negation of negative and negation of positive scores have been excluded from further analysis, because positive and negative scores closely add up to 100% even without these two types. This exclusion does not discourage the use of a specific political sentiment dictionary, because the remaining main types, positive and negative sentiment scores are still informative in describing the tone of the speeches.

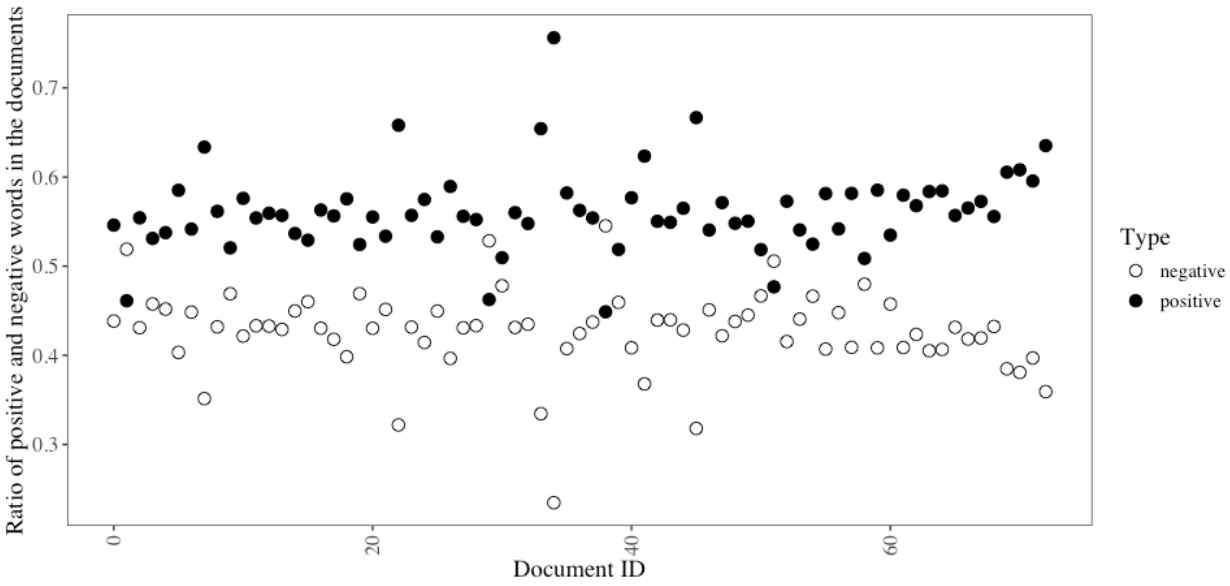


*Figure 6: The ratio of positive and negative words in each document, following a chronological order*

The sentiment scores of Donald Trump's speeches vary more than the lexical diversity, or even the political scaling scores. In case of long documents, such as in the current situation, sentiment score values tend to converge towards neutrality, therefore it is advised, to calculate sentiment scores for bigrams, or sentences, when calculating sentiment for one document. However, in my research I compare document to document, which requires the research to have the overall, document-wide sentiments calculated. Keeping in mind the convergency towards neutrality, the sentiment scores of the speeches indicate overall positive speeches, because the ratio of politically positive words are in most cases higher than 0.5.

Two main takeaways from the general sentiment scores are the spike at and around the 35$^{th}$ speech and the increasing trend of the last 10 speeches. The 35$^{th}$ speech was given on the 5$^{th}$ of October, in Butler, Pennsylvania, the exact location where the candidate got shot on the 13$^{th}$ of July. Based on the speech, he made a point to response to the assassination attempt with kindness, resulting in an overly positive speech. Moreover, this was the first rally where Elon Musk accompanied Donald Trump, giving a speech at the podium. These two facts altogether ensued the high ratio of positive words. Moreover, this speech is accredited for the second highest political scale score (-0.106), implying, that a more positive speech would be categorized more of a neutral speech, rather than moderately or strongly republican. Furthermore, an increasing trend of positive labeled tokens can be detected when analyzing the last 10 speeches, which were given in the span

of roughly 4 days, the last 4 days of the election campaign. The general sentiment of the publicity and most of the polls around that time were favoring Trump as the possible winner of the elections. The increasing positive sentiment might reflect this, because there is currently no evidence supporting the theory, that campaign speeches nearing the end of a long campaign period have significant shift towards positive (or negative) sentiment (Haselmayer & Jenny, 2017).

# III. Statistical modelling

The initial question of my research is whether the statistical characteristics of political speeches have significant relevance to public poll and survey results, or vice versa, narrowing it down to Donald Trump's 2024 election campaign speeches. To understand the connection between the previously calculated statistical properties and public poll results, I had to carefully choose a statistical modelling method focusing on interpretability, for predicting the sentiment of the speeches and the popularity of Donald Trump based on the statistical properties calculated in *Section II.4.1, II.4.2, II.4.3. and II.5*. I expect that the interpretation of the predictive modelling, whether certain properties are important for predicting sentiment scores or popularity or not, will provide an answer for the initial question of this research.

## III.1. Statistical modelling methodology

The two main groups of machine learning models are supervised and unsupervised learning algorithms. Supervised refers to the algorithms where the outcome variable ($\widehat{Y}$) is known in the training dataset, which in this case means either sentiment scores, or polling results. (Rothman, 2018). Therefore, the best-suiting model shall be from the family of supervised machine learning models. The possible outcome variables, positive sentiment score and popularity are both continuous variables, thus the chosen model must be of regression-type.

### III.1.1. Linear regression

Linear regression is often the first choice when it comes to interpretable statistical modelling, however in this section I would like to point out several disadvantages of using linear regression (Salih & Wang, 2024). The equation below represents a simple linear regression equation assuming *k* variables.

$$Y = \beta_0 + x_1\beta_1 + x_2\beta_2 + \cdots + x_k\beta_k + \varepsilon$$

Linear regressions are one of the easiest to interpret, because the p-values for the partial t-tests on the coefficients, are trivial to calculate. Accepting H0 implies that the coefficient has no relevance in the model. Moreover, $\hat{\beta}$ coefficients are not only feature relevance indicators, but they are also the individual feature importance scores, if all predictor variables are of the same measurement unit, enabling advanced analysis on the effect of chosen variables, for instance statistical path analysis. However, linear regressions have numerous disadvantages. It assumes a strictly linear relationship between the predictor variables and the outcome variable, which is rarely the case. Linear models also assume no interactions between predictor variables, resulting in misleading or biased variable interpretation. Multicollinearity, the phenomena when several predictor variables are highly correlated and partially co-dependent on each other, decreases the reliability of the variable p-values, because the model calculations will assign abnormally high standard errors for the involved variables. This directly leads to biased, enlarged p-values and therefore insufficient variable importance measures. All the above-mentioned problems with linear regression are addressable with competent model specification methods, however a highly specified model might lead to significantly decreased interpretability while going from a simple model to a mathematically much more complex model. The equation below is an example of a complex, nonlinear regression assuming *k* variables (Wooldridge, 2012).

$$Y = \beta_0 + x_1\beta_1 + x_2^2\beta_2 + e^{x_3^3}\beta_3 + \ln(x_4^4)\,\beta_4 + x_1x_2\beta_5 + \cdots + x_k\beta_k$$

### III.1.2. Random forest regression

The other possible regression method for the research, due to computational and interpretability reasons is random forest regression. This approach eliminates most of the complexity-related problems of the linear regression models without the need of manual model specification. At the same time, implementing a different, less trivial proposal for individual feature relevance calculations.

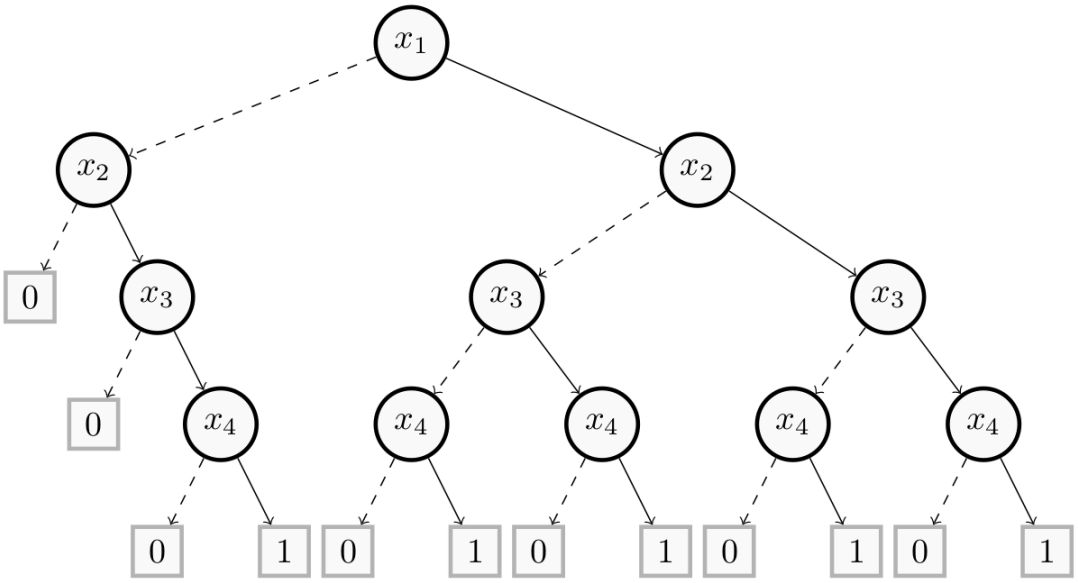


*Figure 7: Representation of a decision tree. Source: PyXAI documentation: Generating models*

Random forest modelling in based on decision trees. A decision tree is a non-parametric supervised learning algorithm, primarily used to solve simple classification problems. It is a flowchart-like structure, splitting the data into smaller and smaller groups (subsets) based on the predictor variables until arriving to a specific decision on the predicted value of the outcome variable. The tree starts with all the data belonging in the same group. A first decision is made on a specific predictor variable, splitting the dataset into two categories. The second decision might be the same for both newly created sub-nodes, and it might differ, fitting the specific sub-set of data. The decision tree recursively splits the data into smaller sets until eventually arriving at a pre-set threshold, or limit of sub-nodes and determines the final class of the data sub-set belonging in the last sub-nodes. The advantage of decision trees is the interpretability and the little computational requirements. On the other hand, disadvantages arise when mentioning the predictive capabilities of this method. Decision trees tend to overfit the training data, meaning that prediction is less accurate for unseen data than it is for the data it has been trained on (Rothman, 2018).

Random forest regression aims to keep the positive attributes of decision trees, such as little computational requirements, while offering a solution for the overfitting nature of simple decision trees. It is an ensemble machine learning algorithm, which combines predictions of multiple decision trees to improve prediction accuracy and reduce overfitting. The dataset is

separated into bootstrap samples. Bootstrap sampling (also known as bagging) means that taking IID (independent and identically distributed) samples from the original dataset. Hyperparameters are model settings which can be altered to fit different needs. Random forest regression's hyperparameters include the number of decision trees, required minimum amount of data on each leaf after a node split, or the maximum number of node splits, also known as the depth of the decision trees. A typical random forest algorithm operates with around 300-500 separate bootstrapped datasets, however this hyperparameter can be adjusted based on different needs, depending on how closely we want the model to follow the outcome variable. For each bootstrap dataset the algorithm separately trains a simple decision tree, with the difference, that each decision tree is assigned with $k$ randomly selected predictor variables ($k < n$, where $n$ equals the number of predictor variables), and each decision, also known as node-split is based only on one of the $k$ predictor variables. It follows that other predictor variables are not relevant. After each of the decision trees are trained, all the total training data is run through all decision trees, and the random forest algorithm takes the average or weighted average of all decision tree outputs to make the final prediction. In case of a continuous outcome variable, the decision tree output is not a class or category, but a single number, which would be very misleading when using only one decision tree. However, random forest regression calculates the average of all decision tree outputs while determining the prediction for a specific datapoint (Behesti, 2022), (Breiman, 2001).

$$\hat{y} = \frac{1}{B}\sum_{b=1}^{B} f_b(x^*)$$

This is the equation for predicting a continuous outcome variable for a new data point, where B means the number of bootstrap datasets and $f_b$ the predicted outcome for the new data point at a given $b$ decision tree.

### III.1.3. Shapely Additive Explanations

The individual feature importance in case of random forest regression modelling means that how different predictor variables individually, partially influenced the predicted value for the outcome variable. The Shapely value is originally a method in game theory to fairly distribute the earnings of a multi-player group-based game where everyone is rewarded based on their contribution to the results. This method for calculating a single players individual contribution assumes, that if the game is played without the player in scope with all possible combinations of

the remaining players, the sum of the contributions will give a fair representations of the player's marginal contribution. The Shapley value (Φ) equation for a chosen player (j) in a game (f) is the following.

$$\Phi_j(f) = \sum_{S \subseteq N\{j\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!}[f(S \cup \{j\}) - f(S)]$$

However, this technique is widely used in interpretable machine learning, especially in tree-based modelling, because values can be interpreted as the individual marginal feature contribution to the final predictions. SHAP is the optimized version of Shapley value calculation for tree-based machine learning models, to make the calculations more computationally effective. SHAP stands for Shapley Additive Explanations and the main advantage is that SHAP values are – as the abbreviation suggests - additive. Summing all feature's SHAP value and the expected value for the outcome variable will result in the value of the prediction (Dr. Kübler, 2024). Take a random forest model built for predicting salary based on age, education and gender as an example. The individual SHAP value of *person1* for age, education and gender added together will give the difference between the average salary of the dataset and the predicted salary of *person1*.

### III.1.4. Regression model evaluation

To evaluate a regression-type model the most used metrics are mean squared error and $R^2$. Mean squared error calculates the total errors of the predictions meaning difference of the predicted outcome variable value and the real value. The calculated error gets squared for all predictions, and the method calculates the mean of these squared errors, to be more robust and handle occasional large errors better. The MSE is not represented on a scale, therefore there are no agreed 'high' and 'low' values, it depends on the predictor variables. Thus, MSE value analysis is not sufficient for evaluating a single model. However, it is a powerful method for the comparison of two or more similar models, similar approaches (Foster, et al., 2014).

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

On the other hand, $R^2$, or the coefficient of determination uses a very similar approach. The main difference is that while MSE is not measured on an exact scale, $R^2$ takes a value between

0 and 1. Technically it is possible to reach a score below zero, however it means that the model's predictions are worse than if the model used only the train data empirical mean for prediction. Generally, $R^2$ is calculated as the quotient of the sum of squared residuals (errors), and the sum of squared total, which is the numerator for the target variable's variance. This calculation ensures that the score does not go over 1, because it is theoretically only reachable if all predictions fit perfectly, and all the error terms are zero. In statistics, evaluation score between 0.1 and 0.5 is considered a medium accuracy, above that there is a strong correlation between the predictor variables and the outcome variable (Foster, et al., 2014).

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}$$

Modell evaluation is usually calculated for test sets. For modelling purposes, the data is split into train and test sub-sets, where train sets are used for training the model and test sets are used for evaluating the models predictive ability. Overfitting means that the evaluation metrics receive significantly better score on the train set than on the test set, meaning that the model predictions are not accurate on unseen data (Rothman, 2018). Normally, when building a machine learning model on any kind of data, it is advised to focus on results on the test set, because the general purpose of modelling is either forecasting or to be able to utilize the model on new, previously unseen datasets. However, in the case of this research, the purpose for building a model is not for further predictions, but to receive important data from analyzing the model's parameters. Therefore, this research analyzes both test and train datasets in the following chapters.

### III.1.5. Word embedding

Word embedding is a powerful natural language processing (NLP) technique to represent words in a numerical format. Retrieving numerical information from words or sentences is crucial in numerous machine learning applications, therefore simple methods like one-hot encoding do not satisfy the requirements for retrieving important information from words. One-hot-encoding places words in a defined order and assigns an $n$ dimensional vector ($n$ = number of different words) to each, in which all values are zero except for $\bar{v}_i$ where $i$ is the index of the word in the in the defined order. See *Table 5* for one-hot-encoding example (Park, 2020).

| *Word* | *Apple* | *Banana* | *Dog* |
|---|---|---|---|
| *One-hot-encoding* | 1 0 0 | 0 1 0 | 0 0 1 |

*Table 5: One-hot-encoding example table*

Word embedding is based on high dimensional vector representations of words. Each word in a corpus is assigned with a fixed dimensional dense vector on the $\mathbb{R}^n$ vector space, where *n* is the number of dimensions. This type of representation puts all words in an *n* dimensional continuous vector space, enabling calculations regarding the distance between points in the space, as words with similar semantical meaning will be closer in the vector space. *Figure 8* depicts a 3-dimensional vector space representation for the words: wolf, dog, cat, apple, banana. Similar words are closer to each other. Thus, embedding words enables retrieving numerical information more effectively from different documents.

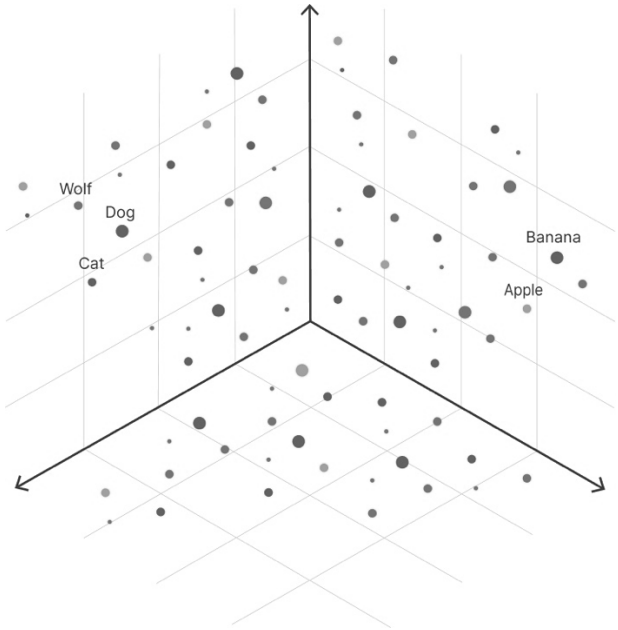


*Figure 8: 3-dimensional vector space representation. Source https://weaviate.io/blog/vector-embeddings-explained*

This type of word representation is typically provided through unsupervised learning techniques, where the entire corpus is given as learning data, for receiving more accurate embeddings. The embedding vectors are usually considered as the output vectors of a shallow neural network. These neural networks are trained using two widely used NLP techniques, the skip-gram method and the continuous bag of words method. To put it simply, the objective of the skip-gram method is to predict the surrounding set context window of an input word, while the

continuous bag of words method is its inverse, where the objective is to put the central word of a given context (Park, 2020).

## III.2. Modelling framework

### III.2.1. The dataset used for random forest regression

When working with small datasets, data quality is a crucial part of modelling. Creating a noise-proof, clear dataset increases the modelling accuracy. In the case of this research, despite the length of the texts, analyzing a total of 73 speeches is small, therefore I had to take pre-cautions to leverage as clean data as possible.

As I previously mentioned, the final dataset contains 73 speeches. The dataset used for random forest regression is based on the same number of speeches, enhanced with the mentioned and calculated statistical characteristics of the speeches in *Section II.2, Section II.4 and Section II.5*. For each speech I have collected the positive, negative, the lexical diversity score, and where the speech is placed on the republican-democrat political scale. At first it might seem reasonable to handle this dataset as panel data, however, for the sake of interpretability I decided to remove all direct indicators of a time series dataset, therefore the timestamp when the speeches took place are not included. However, to account for the lost explainable variance, several variables describing seasonal patterns were added to the model. The month and day of the month were added separately, as well as on which day of the week the speech was given on. Moreover, I calculated the frequency of speeches at a given day, meaning how many speeches did Donald Trump gave on the previous 7 days. The state where a particular speech was held on also takes part in the modelling dataset, as well as the number of previous speeches given in that exact state.

One of the most important variables of the modelling dataset is the popularity of Donald Trump on each occasion when the speeches were given. It is a well-known fact that certain survey's, or organizations behind survey's are favoring certain candidates. While all poll results might statistically be correct, it is not advised, to rely on one chosen survey for comprehensive research like this (Jackson, 2016). Therefore, to retrieve the popularity of Donald Trump on each day when an election campaign speech was given, I have calculated the average of available poll results based on 270ToWin's aggregation, retrieved on February 3rd, 2025 (270ToWin, 2024).

The last component of the modelling dataset were the vector representations of certain words. As previously mentioned, working with the most frequent words is not necessarily the best solution, because it is likely, that those are not the words defying the corpus, or documents of the corpus. However, the TF-IDF scores are better representations of the documents. Studies have shown, that using merely the TF-IDF scores could be an accurate representation of certain words in statistical modelling or analysis (Zhao, et al., 2018). In the case of this research, I have decided to take it one step further and choose the 5 words in each document with the highest TF-IDF score and include their vector embedding representations in the final modelling dataset as predictor variables. For this purpose, the widely used Word2Vec embedding (Mikolov, et al., 2013) generator model has been trained on the entire corpus, creating dense vector representations for each word, where semantically similar words are closer to each other in the n dimensional vector space. However, due to the necessary dimension reduction processes only the mean values of the dense vectors are in the final modelling dataset.

A total number of 17 columns are present in the modelling dataset, making it a 17x73 matrix without missing values. All values have been calculated carefully and as accurately as possible, enabling precise further modelling.

| *Name of the variable* | *Name in the code* | *Description* | *Scale* |
|---|---|---|---|
| Document ID | doc_id | The ID of the speeches | Text1-text73 |
| Month | month | The month in which the speech was held | 6-11 |
| Day | day | The day of the month on which the speech was held | 1-31 |
| Day of the week | day_of_the_week | The day on which the speech was held on | 1-7 |
| State | state | The state in which the speech was held | See *Figure 1* |
| Nth speech in state | nth_speech_in_state | The number of previous speeches given in the state + 1 | See *Figure 1* |
| Negative sentiment | negative | Relative frequency of negative words | 0-1 |

| | | | |
|---|---|---|---|
| Positive sentiment * | positive | Relative frequency of positive words | 0-1 |
| CTTR | CTTR | Corrected type-to-token ratio | See *Section II.4.1.* |
| Scale | scale | Place on the republican-democtrat scale | -1 - 1 |
| Popularity ** | popularity | The calculated popularity on the day of the speech | 0-1 |
| Frequency | frequency | The number of speeches given previously on the same week | 0-15 |
| Vector 1 mean | vector1_mean | The mean of the dense vector representing the highest TF-IDF scored word for the speech | See *Section III.1.4* |
| Vector 2 mean | vector2_mean | The mean of the dense vector representing the second highest TF-IDF scored word for the speech | See *Section III.1.4* |
| Vector 3 mean | vector3_mean | The mean of the dense vector representing the third highest TF-IDF scored word for the speech | See *Section III.1.4* |
| Vector 4 mean | vector4_mean | The mean of the dense vector representing the fourth highest TF-IDF scored word for the speech | See *Section III.1.4* |
| Vector 5 mean | vector5_mean | The mean of the dense vector representing the fifth highest TF-IDF scored word for the speech | See *Section III.1.4* |

*Table 6: Variables used in modelling*

*outcome variable for the model built for predicting positive sentiment

**outcome variable for the model built for predicting popularity

### III.2.2. Random forest regression model preparation

If my initial hypothesis is correct, and the statistical characteristics of the speeches affected the popularity of Donald Trump during the 2024 election campaign rallies, then the said characteristics, or at least a sub-set of them shall be significant when building the models for predicting positive sentiment score and popularity. In this case I mean, that if the popularity is taken as a predictor variable, and one of the statistical properties, for instance lexical diversity is chosen as outcome variable, the individual feature importance of most of the predictor variables should still be significant. However, if there is no connection between popularity and statistical speech characteristics whatsoever, then all combination of predictor variable sub-sets and outcome variables shall indicate little to no individual feature importance (Mentch & Hooker, 2016).

As seen across *Section II*, some of the statistical variables (lexical diversity, corrected lexical diversity, and score on the political scale) have very little variance. These are not used as predictors in our models since inaccurate weights can be estimated for them in our supervised learning model, hence misleading predictions when the outcome variables are not widespread enough. After eliminating the possibly unfit variables of becoming outcome variables, two possible cases remain for modelling purposes. One approach is to have the sentiment as outcome variable and all others as predictor variables, where individual feature relevance will indicate which attributes or metrics determine most of the sentiment scores. If the popularity is among the top variables, it implies a connection between poll results and sentiment, meaning that the sentiment of the speeches is affected by the popularity, for instance a speech is more positive because the popularity is lower, and he is trying to win back voters. The other approach is to place popularity as the outcome variable and all others as predictor variables, where individual variable effect determines whether there is a connection between text characteristics and popularity. If there is, it means that the popularity is affected by the outcome variables, for instance his popularity got higher because his speeches were more positive and more republican leaning. One approach examines whether popularity has affected the tone of the speeches, the other approach examines the opposite, whether the tone of the speeches affected the popularity.

## III.3. Random forest modelling results

### III.3.1. Predicting sentiment

The first model includes positive sentiment score as outcome variable. The results would have been very similar if the negative sentiment were used, due to the small number of neg-negative and neg-positive words positive and negative ratios almost make up to 100%.

For this exact model I included the following variables as predictor variables:

- CTTR
- Scale
- Vector 1 mean, Vector 2 mean, Vector 3 mean, Vector 4 mean, Vector 5 mean
- State
- Number of previous speeches in the same state
- Month
- Day of the week
- Day of the month
- Frequency
- Popularity

For detailed information see *Table 6***.** As per hyperparameters, I trained the model on 400 decision trees with the random state of 2024. The modelling data have been split into train and test sub-data with 0.7-0.3 ratio.

The model evaluation results are in the following table, including the mean squared error and the coefficient of determination for both train and test dataset.

| | MSE | $\mathbf{R^2}$ |
|---|---|---|
| Train set | 0.00028 | 0.8487 |
| Test set | 0.00232 | 0.1359 |

*Table 7: Evaluation metrics of the first model*

The coefficient of the determination ($R^2$) is over 0.6 when calculated for the train dataset, when calculated for the train set, however it is relatively low on the test set. A couple of conclusions could be drawn from the results. It is a typical and relatively extreme case of overfitting, when the model accurately predicts the outcome variables of the train set but has little to no success when attempting to predict for unseen data. My assumption is, that in the case of this research, when the amount of data is considered small ($n$ = 73) and the model by default heavily overfits, the random forest memorizes outcome variables instead of detecting patterns. This means, that even if there are patterns in the data – which is the aim of this research to find – the random forest model's robustness at 400 decision trees generalizes too well in this case, and the patters remain hidden. For such situations two main approaches are available. If there are patterns to discover, the extreme overfitting is either the fault of the model or the data it is trained on. To resolve the overfitting, I fine-tuned the hyperparameters of the model.

If hyperparameter tuning does not fix the overfitting problem, then it is expected to have an inconsistency or other quality related problem with the modelling dataset. To test this, the model for predicting the positive sentiment score have been tested on multiple type of hyperparameter settings. The following table contains the evaluation for a random forest regression model which encompasses 100 decision trees, maximum 5 node splits and a minimum of 6 samples on each leaf after a node split. The reduction of decision trees is needed because of the small amount of data, and the other hyperparameters are needed to reduce the possibility of overfitting by disabling the model to make too many node splits on small amount of data.

| | MSE | $\mathbf{R^2}$ |
|---|---|---|
| Train set | 0.00103 | 0.442 |
| Test set | 0.00222 | 0.175 |

*Table 8: Evaluation metrics of the first model after hyperparameter tuning*

The assumption was right, and with the right hyperparameter settings the extreme overfitting decreases to mild overfitting, because the $R^2$ of the train set descends well below 0.5. However, as the test set's $R^2$ increases only slightly, it indicates the lack of peculiar patterns in the dataset and only a slight connection between positive sentiment and the predictor variables. *Figure 9* below shows the SHAP values for the 72$^{nd}$ speech.

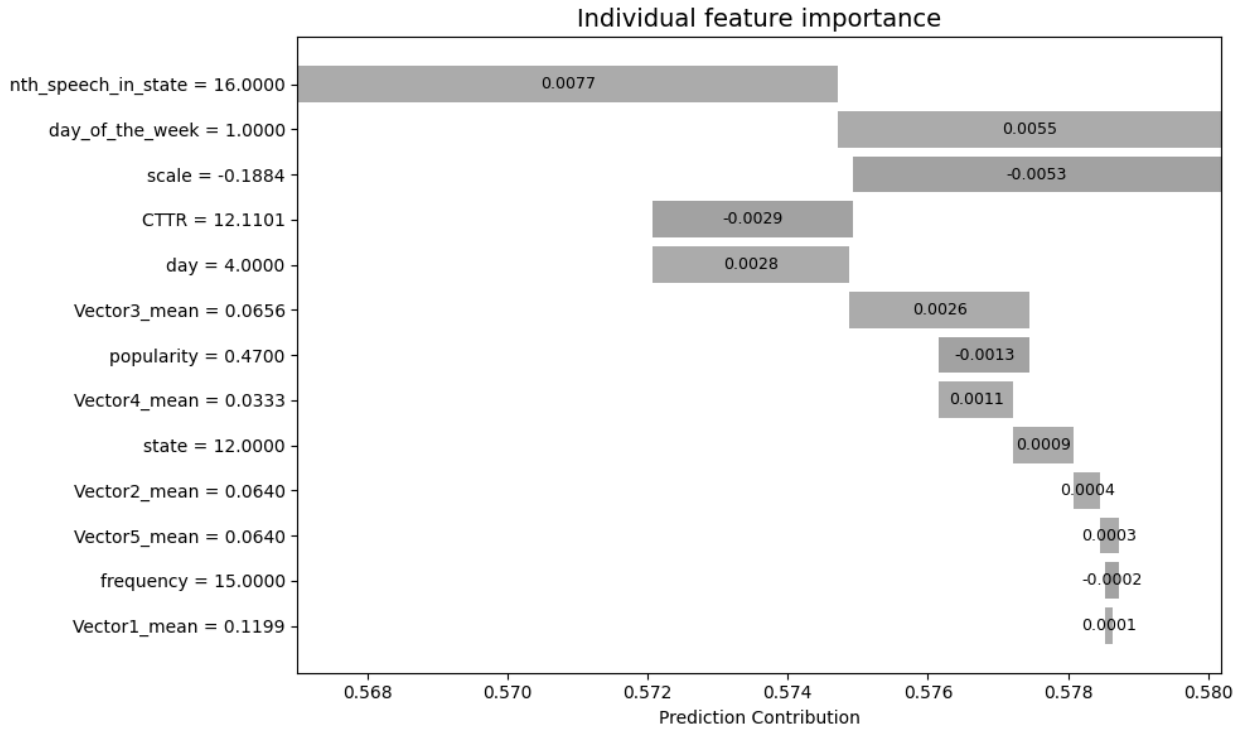


*Figure 9: SHAP values for the 72nd speech*

The original positive sentiment score is 0.595, while the model predicted 0.578, therefore it was slightly below to the actual value. The average of the positive sentiment scores is 0.561, which in this case equivalent to the expected value (*Ef(x)*). Calculating the predicted value is possible via summarizing the SHAP values. The fact, that this speech was Donald Trump's 16$^{th}$ speech in the state (Pennsylvania) increases the positive sentiment score by 0.0077, while the day of the week (Monday) increases it by 0.0055. The third most important feature for predicting the positive sentiment score for this specific speech is the position on the political democrat-republican scale, which decreases the predicted sentiment score by 0.0053. Summarizing the remaining SHAP values with the correct plus and minus signs results in 0.578, the which predicted sentiment score. Important to note, that the popularity is only the seventh most important feature for this speech, and the 5 most important words describing this specific speech have very little importance in predicting the sentiment score. However, it is more beneficial to see the individual feature importance values on *Figure 10* aggregated for all speeches in the train set.

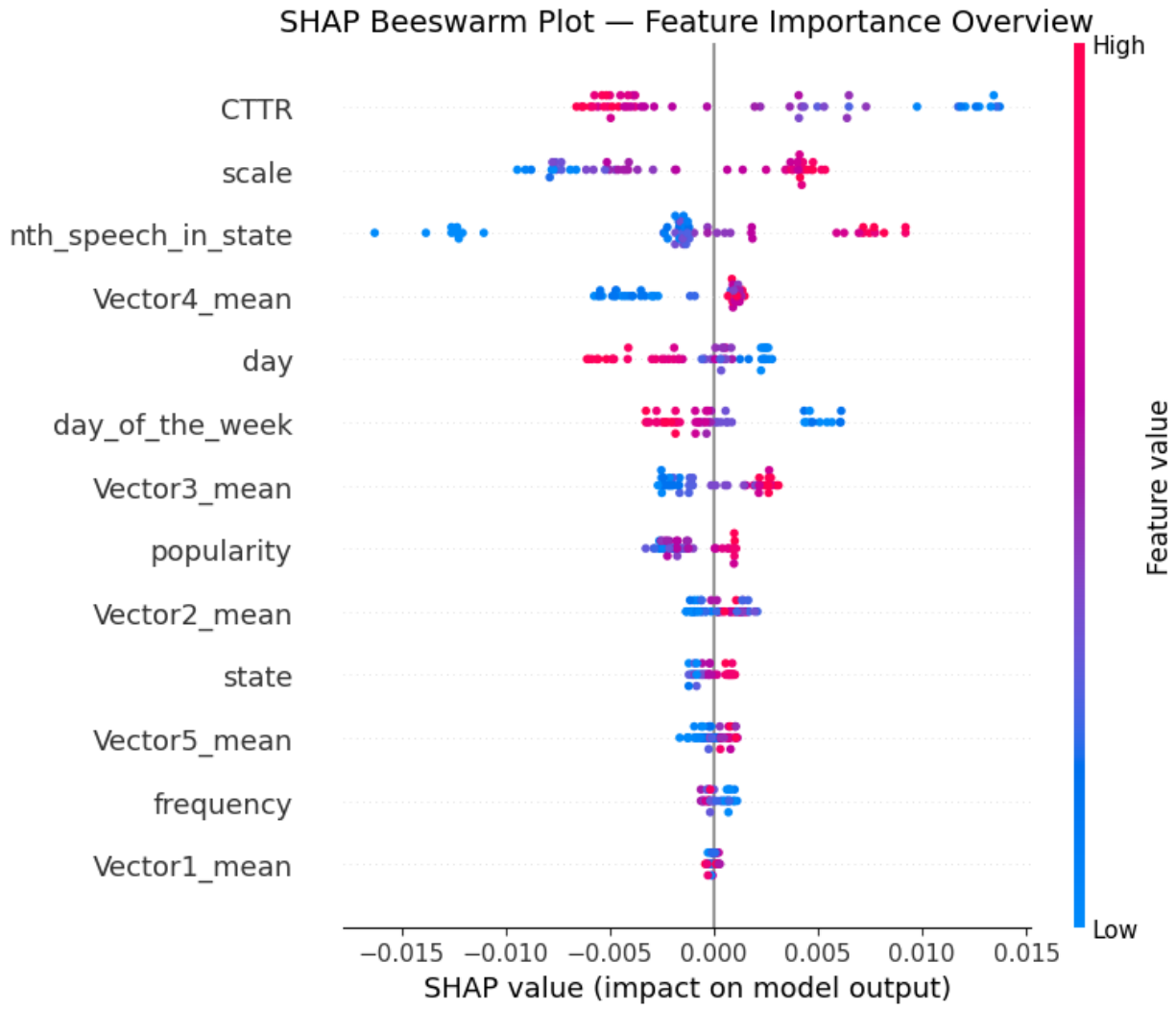


*Figure 10: Aggregated SHAP values*

Based on the beeswarm plot the following conclusions can be drawn. Firstly, the most important features for predicting positive sentiment are the descriptive statistical variables, which is self-explanatory on its own. A lexically more diverse speech might contain more emotions which could draw to a more significant sentiment value, be it on the positive or negative side. In the case of this analysis, a lexically more diverse speech (higher CTTR score) tends to increase the positive sentiment score. Popularity is only the 8$^{th}$ most important feature out of 13, therefore it hardly is significant, let alone an indicator variable when predicting the sentiment of the speeches. However it is important to note, that for most variables the extreme high and extreme low values have significantly higher impact on the model output. In case of CTTR and day of the week variables the very low values have a lot greater positive impact than the mildly low values. Similarly for the

*nth* speech in state feature, the speeches with the lowest number (in this case: 1) have significantly greater negative impact on model output compared to the other low-nth speeches in the state. Such outlier cases indicate that the more extreme a feature is, the greater its impact is on the final prediction, while milder feature values have limited effect on the outcome. This is a parabolical effect, the more a variable is on the edge of its range, the more it affects the positive sentiment.

### III.3.2. Predicting popularity

The other model approached the original hypothesis from a different angle. The outcome variable is the popularity of Donald Trump during the election campaign, while the predictor variables are all statistical characteristics of his speeches. If the fitted model recognizes patterns behind the predictor variables, it could be used to determine the connection between the popularity and the attributes of the speeches.

The predictor variables are the following:

- Day of the week
- State
- Number of previous speeches in the same state
- CTTR
- Scale
- Frequency
- Positive sentiment score
- Vector 1 mean, vector 2 mean, vector 3 mean, vector 4 mean, vector 5 mean

For more information see *Table 6***.** The month and the day of the month predictor variables have been removed, because they heavily influenced the model and the aim of the research is to capture the connection between the popularity and the semantic attributes of the documents, not the meta data of the speeches. Month and day of the month variables affect the popularity, because it brings time series aspects to the model. The month of November influences the popularity because in November Trump had greater popularity than in other months. *Table 9* shows the evaluation of the second model built for predicting popularity.

| | MSE | $\mathbf{R^2}$ |
|---|---|---|
| Train set | 0.00012 | 0.4598 |
| Test set | 0.00019 | 0.0076 |

*Table 9: Evaluation metrics for the second model*

The values are the evaluation of the second model, which has already gone through hyperparameter tuning. The model has been built using 100 decision trees, maximum 5 node splits and minimum 6 sample on each node after a node split. The random state is 2024. The overfitting is even more persistent than at the first model, the coefficient of determination for the test set is hardly greater than 0, indicating, that a similar result could have been reached using only the expected value (*E(y)*) of the popularity. This directly implies, that there are no significant patterns for this specific random forest regression model to recognize and study. Even though the test sets show minimal success at prediction, the train sets were evaluated with an $R^2$ of 0.46, which means regardless of the lack of learnable patterns, some predictor variables show a relationship with the outcome of the predicted popularity in the training set.

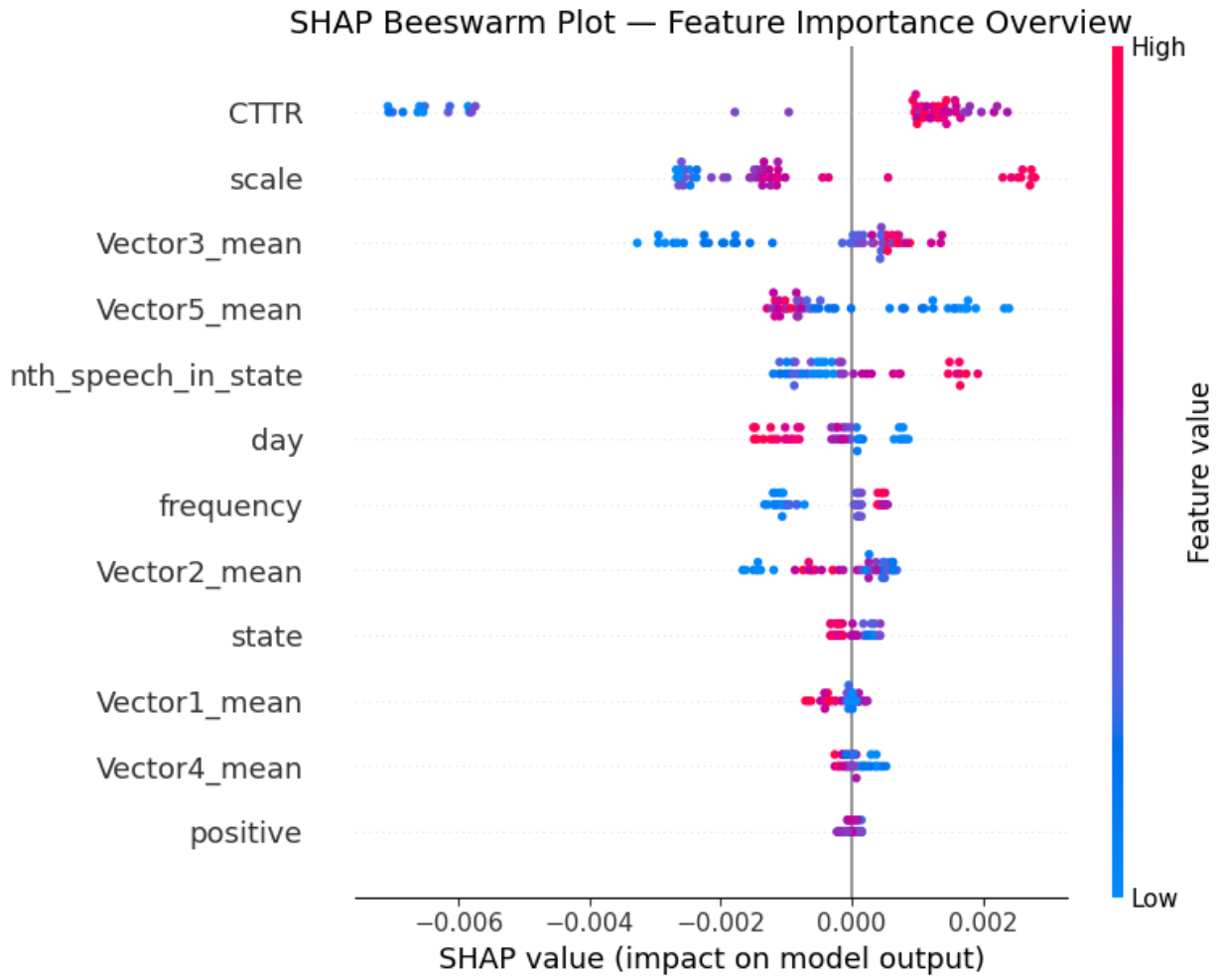


*Figure 11: Aggregated SHAP values*

The beeswarm plot on the train set of the second model is built similar as the plot for the first model. In this case, alike to the model built for positive sentiment score, the corrected lexical diversity is the most important predictor variable. A higher lexical diversity indicates higher popularity, while the low lexical diversity scores and paired with lower popularity. The score on the political scale follows a similar pattern however, it is not clearly divided, that a higher score results in higher popularity, because several above-average scores influence the popularity negatively. The majority of the remaining predictor variables have very little influence on the popularity of Donald Trump on the day when a specific speech was given. As per the previous model, outlier values have exponentially greater impact on model output, especially in the case on corrected lexical diversity, score on the political scale or *nth* speech in state, while not outlier variables are technically insignificant, or close to insignificant.

It is intuitive, to think, that political speeches are either influenced by popularity before the speech's date, or the speech itself influences the coming polls. Therefore, the same models have been constructed encompassing both the popularity on couple of days before and the popularity on couple of days after the speech was given. The popularity on the day afterwards resulted in a slightly better model but with still low explained variance ($R^2$). The evaluation metrics' results are in *Table 10*.

| | MSE | $\mathbf{R^2}$ |
|---|---|---|
| Train set | 0.00011 | 0.4218 |
| Test set | 0.00022 | 0.1433 |

*Table 10: Evaluation metrics for the second model using the next day's popularity*

The popularity on the day before, however, increased the explainable variance even more, as the evaluation metrics' results show in *Table 11*.

| | MSE | $\mathbf{R^2}$ |
|---|---|---|
| Train set | 0.00011 | 0.4349 |
| Test set | 0.00013 | 0.2748 |

*Table 11: Evaluation metrics for the model using the previous day's popularity*

The results show that the popularity on the day before the speech was more influential to the speech than the speech was to the popularity on the day or on the day after the speech. This strengthens the assumptions that there is a slight connection and correlation between characteristics of the speeches and the popularity, but it is unlikely that public political speeches are responsible for voter opinion shifts.

## III.4. Key insights from statistical modelling

After having analyzed the models built for predicting positive sentiment and political popularity, several conclusions can be drawn. Overall, there is only a slight statistical connection between Donald Trump's campaign speeches' statistical characteristics and the related popularity scores. A partial explanation could come purely from a statistical point of view, namely, the number of speeches was simply too small to detect any significant patterns.

On the other hand, the weak statistical connection could at least partially result from the amplification of social media in political opinion forming, and the increasing influence of politicians on popular social media platforms. As multiple studies underline, political right (or republicans) benefit more from social media campaigns, which could directly affect the declining importance of long public political speeches. Moreover, social media with the leading role of X.com has become a major contributor for political opinion forming and for political polarization. Simply, people spend more time reading short political posts on social media, rather than listening to 1.5-2 hour-long live campaign rallies (Balasubramanian, et al., 2024). This suggestion aligns with my modelling results, that only the features with extreme outlier values had significant impact on popularity, because I assume, that more extreme speeches drawn more interest and made greater impact thereafter. Other republican candidates could also amplify the effect of outlier, extreme speeches with consistent mentioning on social media platforms (Huszár , et al., 2021).

This conclusion involves the assumption therefore, that in 2024, most speeches which have influenced the polls could do so, because the more radical speech characteristics reached the publicity easier.

# IV. Conclusion

This present research was concluded to find out the exact impact of the statistical characteristics of Donald Trump's 2024 election campaign speeches on the candidate's popularity. The importance of revealing the connection between political popularity and speeches using statistical analysis and statistical learning methods lays in understanding politicians, their speeches and overall, their motives better, without having to gather information from their campaign teams. Moreover, to question, whether political campaign speeches have a lasting effect on people in the 21$^{st}$ century, with such oversaturated political communication space.

Applied methods include numerous descriptive statistical and text mining techniques which highlighted the characteristics of the speech, including lexical diversity, most important and impactful words, specific political sentiment scores and several others. Furthermore, with the inclusion of popularity as an average of polls by several institutes, I utilized ensemble machine learning modelling to uncover possible hidden patterns and connections between the statistical characteristics of the texts, the popularity and the metadata of the speeches.

The results revealed, that in most cases only minor connections arise between the popularity of the candidate and the given speeches, however, speeches with statistical characteristics that are extreme outliers had significantly greater relevance when it came to individual feature importance analysis. My main argument regarding the reasons for the lack of impact is the oversaturated field of political communication, where long, live-streamed political campaign speeches are less attractive than short-formed social media posts, and citizens are simply more influenced by content which are a lot easier to consume.

However, the question of why official political speeches has little effect on popularity opens new doors and possible research questions, for instance concluding a similar study but incorporating social media data as well, where the statistical methods and results of my analysis offer a solid base.

# Bibliography

270ToWin, 2024. *270ToWin.* [Online]
Available at: https://www.270towin.com/polls/latest-2024-presidential-election-polls/
[Accessed 3 February 2025].

Anon., n.d.

Balasubramanian, A. et al., 2024. A Public Dataset Tracking Social Media Discourse about the 2024 U.S. Presidential Election on Twitter/X.

Behesti, N., 2022. Random Forest Regression. *Towards Data Science.*

Bing, L., 2022. *Sentiment Analysis and Opinion Mining.* s.l.:Synthesis.

Boehmke, B. & Greenwell, B., 2020. *Hands-On Machine Learning with R.* s.l.:Taylor & Francis Group.

Breiman, L., 2001. Random Forests. *Springer Nature: Machine Learning,* pp. 5-32.

Dr. Kübler, R., 2024. Shapley Values Clearly Explained. *Towards Data Science.*

Epres, B., 2025. *GitHub.* [Online]
Available at: https://github.com/brownepres/trump-speech-analysis.git

Foster, L., Diamond, I. & Banton, J., 2014. Correlation and Regression. In: *Beginning Statistics: An Introduction for Social Scientists.* s.l.:SAGE Publications Ltd.

Graham, C. & Roll, N., 2024. Evaluating OpenAI's Whisper ASR: Performance analysis across diverse accents and speaker traits. *ResearchGate,* p. 8.

Haselmayer, M. & Jenny, M., 2017. entiment analysis of political communication: combining a dictionary approach with crowdcoding. *Qual Quant,* Volume 51, p. 2623–2646.

Huszár , F. et al., 2021. Algorithmic Amplification of Politics on Twitter.

Jackson, N., 2016. Chapter 26: The Rise of Poll Aggregation and Election Forecasting. In: *The Oxford Handbook of Polling and Survey Methods.* s.l.:s.n.

Kusnet, D., 2002. *Bush's Inaugural Speech Must Walk the Middle Road,* s.l.: Economic Policy Institute.

Lowe, W., 2008. Understanding Worscores.

Mentch, L. & Hooker, G., 2016. *Quantifying Uncertainty in Random Forests via Confidence Intervals and Hypothesis Tests.* s.l., s.n.

Mikolov, T., Chen, K., Corrado, G. & Dean, J., 2013. *Efficient Estimation of Word Representations in Vector Space.* s.l., s.n.

Ntontis, E. et al., 2023. A warrant for violence? An analysis of Donald Trump's speech before the US Capitol attack. *British Journal of Social Psychology.*

Park, J., 2020. Word Embedding in NLP: One-Hot Encoding and Skip-Gram Neural Network. *Towards Data Science.*

Rothman, D., 2018. *Artificial Intelligence By Example: Develop machine intelligence from scratch using real artificial intelligence use cases.* Birmingham - Mumbai: Packt Publishing.

Salih, M. A. & Wang, Y., 2024. Are Linear Regression Models White Box and Interpretable?.

Silge, J. & Robinson, D., 2024. 5. Converting to and from non-tidy formats. In: *Text mining with R: A Tidy Approach.* s.l.:s.n.

Silge, J. & Robinson, D., 2025. Analyzing word and document frequency: tf-idf. In: *Text Mining with R: A Tidy Approach.* s.l.:s.n.

Sitara, N., 2023. Reflections on President Kennedy's "Strategy of Peace" Speech. *Harvar Kennedy School, BELFER CENTER.*

Torruella , J. & Capsada, R., 2013. Lexical Statistics and Tipological Structures: A Measure of Lexical Richness. *Procedia - Social and Behavioral Sciences,* pp. 95:447-454.

Wicklin, R., 2018. Ranking US presidents. *SAS Blogs.*

Wooldridge, J. M., 2012. Regression Analysis with Cross-Sectional Data. In: *Introductory Econometrics: A Modern Approach.* s.l.:South-Western CENGAGE Learning.

Young, L. & Soroka, S., 2012. Affective News: The Automated Coding of Sentiment in Political Texts.. *Political Communication.*

Zahrotun, L., 2016. Comparison Jaccard similarity, Cosine Similarity and Combined Both of the Data Clustering With Shared Nearest Neighbor Method. *Computer Engineering and Applications Vol. 5, No. 1.*

Zhao, G., Liu, Y., Zhang, W. & Wang, Y., 2018. TFIDF based Feature Words Extraction and Topic Modeling for Short Text.