# Introduction

Following the management team’s interest in our model to predict stock market price using NLP techniques on news articles, they have shown an interest in the training of a model that is able to improve company performance using text data. To achieve this our team analyzed the contents of Nvidia related news articles and identify the topics that are present in these news articles.

To achieve this topic modelling was used for this analysis. By applying topic modelling techniques, we intend to uncover the content of the Nvidia related articles. This report relays the insights gained, inner workings and results of using three distinct topic modelling algorithms, namely Latent Dirichlet Allocation (LDA), Formal Latent Semantic Analysis Weighted (FLSA-W), and BERTopic.

In the following sections the technicalities of these algorithms will be explained, and an overview of their performance will be given. Additionally, the topic modelling outcomes under various environmental changes will also be presented. Finally, the limitations of these algorithms and the manner in which our findings could be applied to real-world financial companies will be discussed.

# The Topic Modelling Algorithms

This section explains how each of the topic models function, extract useful information from the Nvidia news articles along with their weaknesses and strengths. In knowing how these algorithms operate we can better understand their performance and how to improve them.

Latent Dirichlet Allocation (LDA)

According to a classic representation theorem by (De Finetti, 2017) any collection of exchangeable random variables has a representation as a mixture distribution—in general an infinite mixture. It is upon this theory that Latent Dirichlet Allocation (LDA) is based (David M. Blei et al., 2003). LDA is a generative statistical model that is capable of representing words and documents in a way that acknowledges their exchangeability, meaning the probability of encountering a set of words is the same regardless of their order. In other words it assumes that documents are a mixture of topics and topics are a mixture of words. Due to this capability, LDA can be utilized to determine the topics that are present in a large collection of documents. It achieves this by assigning each word in each document to a random topic and iteratively refines these assignments based on the probabilities of word occurrences in topics and topics in documents.

The strengths of LDA lie in how easy its results are to understand and interpret. LDA’s efficiency with large datasets along with its flexibility, through the incorporation of hyperparameters make it an excellent choice for the task at hand which is topic modelling. On the other hand, LDA is context ignorant, meaning it assumes that words are independent of one another(David M. Blei et al., 2003). Furthermore, LDA’s performance heavily relies on the tuning of its hyperparameters. And lastly, it may struggle with short texts. Despite of these shortcomings, it strengths make LDA a solid contender for our topic models.

Fuzzy Latent Semantic Analysis Weighted (FLSA-W)

FLSA uses singular value decomposition to decompose document representations into three matrices UΣVT. It then applies importance weights (like TF-IDF) to this singular value decomposition of the term-document matrix(Rijcken et al., n.d.). This application of importance allows the model to prioritize more significant terms that occur less frequently across our corpus, but are more informative for our context, which in our case would be the Nvidia related terms. Due to this mechanism FLSA-W can effectively highlight differentiated topics and semantic relationships within Nvidia-related content by weighing terms according to their importance.

One of the traits that makes FLSA-W a prime candidate for topic modelling is that its one of the fastest and efficient topic models, due to its computational efficient manner of applying matrix operations. It is also capable of reducing noise on the most significant singular vectors and is excellent at handling synonyms. Unfortunately FLSA-W’s Dimensionality reduction might lead to the loss of important information and it’s more limited in comparison to other models when handling larger datasets.

## BERTopic

BERTopic leverages state-of-the-art transformer models to create dense vector representations of texts. It uses these embeddings, created from the BERT (embedder) to find clusters of documents, which represent the topics(Grootendorst, 2022). Unlike LDA, BERTopic considers the context and semantics of words in texts due to the transformer architecture, leading to more contextually relevant topics. It is context aware, can adjust the number of topics dynamically and performs well even on short documents(Egger & Yu, 2022). However, this prestige comes at the price of it being the most computational intense algorithms we’ve tested. It also sacrifices interpretability for performance, due to its black-box nature. Lastly this model relies heavily on the quality of the pre-trained BERT embeddings.

BERTopic leverages transformer-based embeddings, clustering techniques, and a class-based variant of TF-IDF to generate coherent topic representations. It uses pre-trained transformer models, such as Sentence-BERT, to convert documents into vector representations. These embeddings capture the semantic content of the text.

Since embeddings are high-dimensional, BERTopic reduces their dimensionality using techniques like UMAP (Uniform Manifold Approximation and Projection) or PCA (Principal Component Analysis) to optimize clustering. These methods hep in preserving the structure of the data while reducing the number of dimensions. Once the embeddings are reduced in dimensionality, a clustering algorithm is applied to group similar documents together. BERTopic typically uses HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) for this purpose.

HDBSCAN identifies clusters of varying densities, allowing it to find meaningful groupings of documents while also handling noise (outliers) effectively. Each cluster represents a group of documents that share similar topics. After clustering, BERTopic generates topic representations using a class-based variation of TF-IDF (Term Frequency-Inverse Document Frequency). This approach allows for the extraction of coherent and representative terms for each topic.

For each cluster, the model calculates the TF-IDF scores for the words within that cluster. The class-based approach means that the model considers the cluster as a class and identifies the most significant terms that characterize the documents in that cluster.

BERTopic provides tools for visualizing the topics generated. This can include visual representations such as word clouds, bar charts of the most significant terms, or interactive visualizations that allow users to explore the relationships between topics. We will talk about the topic similarity

Having generated topic embeddings, through both c-TF-IDF and embeddings, we created a similarity matrix by running the (.visualize\_heatmap()) method, which simply applies the cosine similarities through those topic embeddings. The resulted in a matrix indicating how similar certain topics are to each other.

# Analysis

In this section, we will discuss the training of the selected models. Our approach consists iteratively training our topic models and slightly adapting the environment in which they are trained for every iteration. Not only does this improve our models over time but also gives us insights into how different modifications to our input data affects the model’s performance. The decisions that were made, and the setup and changes that were applied to each iteration will also be presented in this section. To improve comparison the same iteration steps were applied to each model’s iteration.

## Iteration 0

This iteration was named ‘iteration 0’, due to us applying some basic preprocessing to the data. After filtering the out all of the Nvidia related articles, the hyperlinks were removed along with the numbers and non-alpha words. Lastly, all of the document text were represented in lowercase. Subsequently, after applying these preprocessing steps we trained the models with their default values.

## Iteration 1

For the First iterations we removed stop words (e.g.  a, the, and , or , of ) from the Nvidia related articles. Using the results gained from iteration 0 we

## Iteration 2

## Iteration 3

# Evaluation and Results

In this section we will assess the performance of the models after each iteration. Each model was assessed qualitatively and quantitatively. For the quantitative assessment we chose to 3 metrics to compare the models. We use the coherence, diversity and interpretability score calculated by utilizing FLSA-W’s (python package called flsaw) built-in functions. The *Coherence Score* measures how well each topic’s words support each other(Rijcken et al., n.d.). A measure for inter-topic quality is topic *Diversity*, which measures the unique words in a topic model as a proportion to the total number of words (Dieng et al., 2020). The *Interpretability Score* combines intra- and inter-topic quality by taking the product between the coherence and diversity score (see the equation below)(Dieng et al., 2020). These metrics were chosen since they allow for us to objectively compare the intra-, inter-topic- and overall quality for each of our models.

*Eq. Interpretability = Coherence × Diversity*

## Qualitative Assessment and Quantitative Assessment