GRAPH-BASED WEBSITE EMBEDDINGS TO OPTIMIZE DIGITAL AD CAMPAIGNS

VYOM PANKAJKUMAR BHATT

A thesis submitted in fulfilment of the
requirements for the award of the degree of
MASTER OF SCIENCE IN DATA SCIENCE

LIVERPOOL JOHN MOORES UNIVERSITY (LJMU)

JULY 2020

ABSTRACT

Neural networks have seen applications in a variety of domains. This study explores how it's application can be extended to one of the less experimented domains, digital advertising. Digital ad campaigns can be optimized in several ways, from contextual targeting to cookie-based targeting strategies. This study delves into the application of geometric deep learning techniques to optimize digital ad campaigns contextually, by identifying the relevant websites for ad targeting. This work proposes using a graph-based representation to identify connections between different websites in terms of the common users shared. This is followed by using researched deep learning techniques to represent these websites in a low-dimensional vector space by generating node embeddings. These embeddings would be representative of the relationship that exists between the websites. Eventually, these embeddings are clustered to form segments of websites. A custom evaluation criterion is designed to identify the ideal set of website clusters that can be used to optimize digital ad campaigns. With the regulations surrounding the use of web cookies for campaign optimization getting stricter, this study aims to fill the increasing demand and gaps existing in the current contextual targeting capabilities.

Key Words: website embeddings, graph embeddings, geometric deep learning, digital advertising, node2vec, website clustering, neural networks, contextual ad targeting

ACKNOWLEDGEMENTS

This thesis has been written in fulfilment of the degree of Master of Science in Data Science at Liverpool John Moores University. I would like to acknowledge the following people, without whose assistance, this thesis would not have been possible. Firstly, my supervisor Mr Bharath Kumar Bolla for his technical advice, constructive feedback, and constant encouragement provided throughout the research period. I would like to thank Dr Manoj Jayabalan from the Department of Computer Science, Liverpool John Moores University, for his guidance on how to structure the thesis and constant motivation. I would like to thank my mentors, Anmol Jain and Bhavika Bhambhani, for their everlasting zeal and constant support. Finally, I would like to thank my parents and my peers for their constant support throughout this research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DSP…….....	Demand-Side Platform
SSP…….....	Supply-Side Platform
GDPR…….....	General Data Protection Regulation
PCA…….....	Principal Component Analysis
t-SNE……..	t-distributed Stochastic Neighbour Embedding
RTB……	Real-Time Bidding
PMP……	Private Marketplace
JSON……..	JavaScript Object Notation
KL Divergence…...	Kullback-Leibler Divergence
DBSCAN…….....	Density-Based Spatial Clustering of Applications with Noise
URL…….....	Uniform Resource Locator
CRISP-DM…….	Cross-Industry Process for Data Mining
HTTP……..	Hypertext Transfer Protocol
BFS…….	Breadth-First Sampling
DFS…….....	Depth-First Sampling
SC……...	Silhouette Coefficient
CE……...	Cluster Error
CCI…….	Cluster Confidence Index
SDNE…….	Structural Deep Network Embedding
LINE……...	Large-scale Information Network Embedding
WO…….	Website Overlap
CS……	Cluster Skew

CHAPTER 1

INTRODUCTION

The digital advertising space has seen massive growth in recent times as the worldwide digital ad spend is expected to exceed $350 billion by the end of 2020, and over 50% of all ad spend would comprise of digital ads (Enberg, 2019). Websites on the internet hold a major part of the ad inventory that is available for advertisers to reach their customers.

The major challenge for any advertiser, however, is identifying websites where they should serve ads to ensure they're reaching their potential customer (Agarwal & Shukla, 2013), hence avoiding wastage of ad spend on irrelevant websites. Marketers need to constantly keep optimizing their digital ad campaigns to ensure they're spending on ad slots on websites that are likely to perform better.

This study proposes a data-driven approach to perform this optimization, which is advertiser vertical agnostic. It first attempts to represent website visitation patterns of a set of users in the form of an undirected graph, with websites as the nodes and edges being the number of users common to both connecting websites. Post that, the study explores the usage of researched geometric deep learning techniques (Grover & Leskovec, 2016) to generate nodal embeddings for the graph structure. Having generated the embeddings, the study aims to generate different clusters of websites that can be used in groups for targeting purposes in digital ad campaigns.

With the share of digital advertising in the advertising landscape increasing ever so quickly, there is an increasing demand for studying and implementing data-driven optimization techniques to improve campaign performance. This study aims to contribute to the domain by bolstering the armoury of techniques that marketers have at their disposal for performing ad campaign optimization. At the same time, the study also explores novel applications of previously researched techniques pertaining to data science in the field of digital advertising.

1.1 Background of the Study

Digital advertising, also known as online advertising or internet advertising, involves showing promotional ads to users on websites they are browsing. The advertisements can take different forms such as image ads, video ads, gif ads, interactive bot ads, etc. and are served to users on websites that hold ad slots/ad inventory.

The ad serving process involves real-time bidding for ad slots, made possible by platforms (commonly referred to as demand-side platforms or DSP's), and provides the advertiser with the flexibility to select the type of inventory to bid on by customizing their targeting strategies on the DSP.

Once a user logs on to any website which has an available ad slot, the bidding process kicks in. If the ad slot characteristics (referred to as contextual characteristics and includes features like the website, the website category, the position of the ad slot on the webpage) and the user characteristics (referred to as audience characteristics and includes features like user demographics, web-cookie based features) match the targeting strategies set by any advertiser on their DSP, then their bid goes through and competes with other advertisers bidding for the same ad slot. The advertiser with the highest bid wins and serves their ad on the ad slot to the user. This entire ad serving process takes a matter of milliseconds.

Like other methods of advertising, the success of digital advertisement depends a lot on how effectively these ads are served i.e. whether these ads are reaching the ideal audience (web user) on the ideal platform (website). Though unlike other methods of advertising, digital advertising has an added advantage with regards to the flexibility it provides when it comes to customizing the type of websites to serve ads on, and the type of users to serve ads to. Advertisers have complete control over the type of websites their ads are begin served on and the demographics of a user on the internet that they're showing the ads.

The major challenge that advertisers face in digital advertising is identifying the websites or audience demographics that are most relevant to their brand and hence are most likely to click through to their websites via the ads. This boils down to an optimization problem for the advertiser as they strive to improve their ad targeting strategies throughout the ad campaign to achieve better performance.

As discussed above, the types of ad targeting that is available to advertisers in the digital advertising domain can be broadly classified into two groups:

- Contextual Targeting: involves the targeting of advertisements on specific websites that have been identified by the advertiser as relevant to their brand.
- Audience Targeting: involves the targeting of advertisements to specific users, having analyzed their demographics by capturing their web cookie data.

Advertisers usually split their ad budgets into these two broad categories and optimize ad spends further based on the performance of each strategy.

However, with governments around the world become more sceptical about advertisers and agencies capturing web cookie data for users on their websites, they have begun introducing stringent data protection laws making it difficult for advertisers to capture cookie data. The GDPR (general data protection regulation) in the European Union and the European Economic Areas is one such law, for example, that has made the holding of personally identifiable information by websites illegal without the consent of users and without a legitimate purpose of holding the data. These regulations, in turn, impact the audience targeting technique discussed above as cookie-based targeting would eventually fade out in a world with data protection acts imposed by governments.

This makes the contextual targeting strategy, and identifying different sophisticated techniques for it, even more important. This study proposes the use of researched geometric deep learning techniques (Grover & Leskovec, 2016) on a graph representing links between websites to propound a novel technique for contextual targeting in digital advertising campaigns.

1.2 Problem Statement

The need to identify sophisticated contextual targeting techniques is imperative as the marketers attempt to cope with stringent data protection regulations imposed by governments as cookie-based campaign targeting becomes redundant, and the existing strategies are not sophisticated or heavily data-driven. The problem at hand is to first procure a dataset with contextual website features, then use it to come up with an analytics-driven approach to produce solutions that can be used for optimizing the targeting strategies in the digital advertising domain.

This study proposes to use different geometric deep learning techniques to generate website embeddings – which is, essentially, representing the websites in a vector space; and use these embeddings to suggest methods of optimizing digital ad campaigns. This study then compares the different deep learning techniques using a custom-defined criterion to identify the most optimal method that serves the purpose.

1.3 Aim and Objectives

This research aims to propose a data-driven approach to the optimization of digital campaigns. The study leverages learnings from previous research in graph-based geometric deep learning techniques to generate embeddings of websites that can be used to optimize digital ad campaigns.

The research objectives are formulated based on the aim as defined above and are as follows:

- To represent the websites in the form of a graph structure.
- To model website embeddings from the graph using geometric deep learning techniques.
- To generate clusters of relevant websites using the embeddings.
- To design a relevant criterion and evaluate the various techniques used in the methodology to identify the best performing technique.

1.4 Scope of the Study

The work done in this study is limited to the data available from web traffic requests for November 2009 (Meiss et al., 2010). Hence, the embeddings generated for the websites would only be reflective of the browsing behaviour of users during the timeframe captured in the dataset.

The aim is to generate website embeddings and evaluate how different embeddings can efficiently cluster the websites. The applications of these website embeddings can be extended to other applications in digital ad campaign optimization, such as:

- predicting the next website that a user would visit based on their web history.
- using the website embeddings in machine learning problems which have websites as one of the predictor variables.

This research does not study these applications in the methodology but explores these two applications later as a scope of work due to the non-accessibility of open-source datasets.

Training of the neural networks is a computationally expensive task, and the number of resources available at hand as part of this research was limited. Due to this, a limited number of hyperparameters have been tested so that the processing resources can handle them.

The embeddings generated are limited to website-website interactions present in the available dataset. This can be extended to user-website interactions, which would intuitively be a fairer representation of the similarities existing between websites. However, user interactions aren't considered as part of this research due to constraints in time and resources.

The evaluation of the methodology is done using a custom criterion that aims to score the clusters generated from the base dataset. This study does not evaluate the clusters of websites formed by quoting a performance uplift on any ad campaign as the scope of the study is limited to academic purposes. However, the methodology implemented here can be extended to any marketing data in a similar format and be used for optimizations and evaluation using marketing performance parameters.

1.5 Significance of the Study

This research contributes to the field of digital advertising as it proposes a data-driven approach for contextually optimizing digital ad campaigns. In the literature review that follows, the existing techniques that are being implemented in the space of digital ad campaign optimization are discussed, along with the disadvantages that currently reside with them. The study proceeds to look at the gaps that exist with these techniques and how this research can be utilized to fill those gaps.

This study further investigates the methods of how this work contributes to the field of advertising using an industry-relevant dataset. The work from this project would be valuable to anyone working or performing research in the digital advertising or ad-tech domain.

From the literature review that will be presented in the next chapter, it is evident that there exist several limitations in techniques used for digital ad campaign optimization. Currently, the techniques are either based out of intuition or are very mildly data-driven. Even in cases where the strategies are data-driven, the learnings from those datasets are very specific to an advertiser, or a vertical, and hence they can't be extended across the landscape. The vertical here refers to a category of advertisers, such as gaming, sports, etc. The technique proposed in this study aims to capture the behaviour of users on the internet, and hence the output strategy can be extended across different verticals or advertisers and is completely supported by data.

The implications of stricter government regulations pertaining to the use of web cookie data pose a serious concern to the current user targeting strategies for digital ad campaigns. As the use of cookies gets deprecated, there would be an increased requirement for other smarter contextual strategies for digital ad targeting. The research done would contribute towards these strategies, as it is robust and not cookie-based, hence providing marketers with a strategy to fall back to when the regulations get stricter.

In addition to contributing to the digital advertising landscape, this project also contributes to the applications researched in the field of geometric deep learning techniques performed on graph structures. Neural network embeddings have historically primarily been used in the space of natural language processing where it has seen massive success with some innovative projects. This study explores a novel application for using neural network embeddings in the space of digital advertising for campaign optimization. Research on this novel approach would add to the studies done in the field of embeddings.

1.6 Structure of the Study

In this chapter, the structure of the entire thesis is discussed to give an overall understanding of the flow of the report. Chapter 1 introduces the topic of this research, the domain that this research pertains to, and a broad idea of the gaps that this research is trying to fill. Chapter 1.2 clearly defines the problem statement that is being addressed, and the following section crisply states the aim and objectives that are the expected outcome from the research. Chapter 1.4 presents the scope of the study, which also includes the limitations that existed during the study. Following that, chapter 1.5 addresses the significance of the study and how it is contributing to the domain.

In chapter 2, there is an exhaustive account of the literature review that was done pertaining to this study. The flow of the entire chapter is such that first, it establishes the background knowledge and related publications regarding the domain that this research belongs to i.e. digital advertising, then it touches upon the gaps that exist in the domain and the motivation behind this study, and finally presents the review done on the technical aspects that form the basis of the methodology that follows.

Chapter 3 discusses the methodology that is followed and the framework that is proposed for the study. Chapter 3.2 presents the methods used to source, clean, and transform the dataset into a desirable graph structure. The chapter proceeds with introducing the node2vec technique used to generate website embeddings from the graph. Post generating the embeddings, chapter 3.4 discusses the dimensionality reduction techniques and their purpose in this research has been

discussed. Chapter 3.5 discusses how the embeddings' dataset is clustered using three different clustering techniques to produce a final set of clusters. The chapter closes with the final objective that needs to be achieved i.e. evaluation of the methodology used. A novel evaluation criterion, designed to evaluate the techniques used in the methodology, is presented.

Chapter 4 presents the exploratory data analysis done on the dataset. This chapter presents the analysis done to understand the nature of the data in both the tabular and the graph structure. The insights generated from this chapter govern the implementation of the model in terms of selecting the architecture that is most suited to the structure of the data.

Having analyzed the data, the 5^{th} chapter presents the results of the models that have been implemented. The evaluation criterion designed for this study is used to identify the best set of models suited to the dataset for achieving the objectives set out. Having identified the best set of models, the results from this set are presented in the context of their application in digital ad campaigns.

Finally, this study closes with chapter 6 that summarizes the work done in this project. A critical review of each of the objectives is presented, along with how they have been achieved throughout this study. There is a summary provided on the major limitations and the scope of this study. Along with that, there is an outline of the contribution made by this study to the existing knowledge in the digital advertising domain and the recommendations on how this work can be taken forward in the future.

CHAPTER 2

LITERATURE REVIEW

In this section, the extensive background research that was done as part of this study on the digital advertising space and its optimization aspects is presented. The chapter also discusses research done on the technical aspects pertaining to this project and the motivation behind using those techniques by evaluating the previous works of researchers. The literature review presented here defines the inspiration behind this project and sets the foundation for the methodology that is presented later in this study.

2.1 Introduction

The topic of this research is based on the digital advertising domain and understanding the nuances of this research would require some prior knowledge of the domain. As part of the background research and literature review, this chapter first explains the concepts pertaining to the digital advertising domain, proceeds to present the existing methodologies in the domain, and points out the gaps existing in the current state to help define the motivation of this study.

The chapter then proceeds to observe the gaps in the current digital advertising scenario by taking an analytical approach which would form the premise of the methodology that follows in the further chapters. The chapter then ventures into presenting the literature review done in the technical data science aspects that are proposed to be implemented in the methodology to solve the problem identified.

In particular, the chapter discusses the research around how neural network embeddings have evolved, beginning from the word2vec embeddings to represent words in a vector space (Mikolov et al., 2013a), to node2vec embeddings to represent connected nodes in a graph on a vector space (Grover & Leskovec, 2016). The chapter introduces the work done previously with graph-based deep learning techniques and establishes its relevance in the domain.

Also, the chapter introduces the dataset that has been used as part of this research and presents the previously researched work on the same. Other potential datasets are discussed that could have been used in place of the one used currently to make a case for the benefits that the current dataset provides concerning this research.

Along with the above review, there is a discussion on the background research done on the data science techniques that have been implemented in this methodology. Concepts regarding dimensionality reductions using PCA and t-SNE, linear and non-linear clustering techniques, hyperparameter tuning in machine learning and deep learning frameworks have been used in the methodology, and this chapter presents an exhaustive literature review done on them.

The chapter concludes by identifying how this study manages to close the gap that exists in the digital advertising domain that was identified. The methodology implemented is data-driven which could potentially act as an inspiration for more research to be done using geometric deep learning techniques as an application in optimizing digital campaigns.

2.2 Fundamentals of Digital Advertising

Digital advertising is one of the many advertising solutions that marketers have at their disposal when it comes to advertising their products. Digital advertising involves using the internet as a medium of advertising rather than conventional mediums such as newspapers, radio, billboards among others (Yasmin et al., 2015). Some of the common forms of digital advertising include advertising using emails, advertising on search engines, social media advertising, display advertising, etc.

The area of interest for this study is primarily display advertising, however, the methodology proposed can be easily extended to other forms of digital advertising. Display advertising refers to the banner advertising that is seen on websites that are clickable and direct users to the website pertaining to the advertisement. The main aim of display advertising is to drive more traffic to an advertiser website and get the users to convert or purchase any product/subscription that the advertiser has to sell on their website (Klein, 2010).

From an advertiser's point of view, the process of advertising over the internet using display ads has evolved. Initially, advertisers who want to serve ads on specific websites used to strike a deal with the website to purchase the inventory. Once the deal has been completed, the inventory or ad slots agreed as part of the deal on the advertiser website would only display ads served by that advertiser.

This form of traditional media buying brought with it several disadvantages. It compromised the flexibility that advertisers have on the users and the content that they want to serve their ads on. That is because a deal binds them to serve ads on the agreed inventory for a fixed amount of time, which means they can't switch inventories to optimize their campaigns better. This form of buying was also not very cost-effective as the prices of the ad slots were fixed and directed by the owner of the inventory. To curb these issues, the ad-buying format has evolved to a more automated format, which is now referred to as programmatic ad buying.

2.3 Programmatic Ad Buying

Traditional ad-buying brought with it several drawbacks as discussed. Additionally, the format was not very robust as the demand for advertising increases. As the demand for ad slots increases, there needed to be a much more evolved structure to process such a large scale of advertisements. This led to the programmatic form of ad buying, the method which is used currently to serve ads over the internet.

Programmatic buying of ad, as the name suggests, is the automated method of buying ad slots on websites over the internet using a real-time bidding system (Choi et al., 2019). This system enables advertisers to purchase ad slots available from a variety of websites in a matter of milliseconds. With the introduction of this method of buying, advertisers no longer need to strike long term deals with websites for serving ads. Programmatic ad buying also gives advertisers the flexibility to switch between websites they are willing to serve ads on at any point in time.

There are several components to the programmatic bidding mechanism. The main components of programmatic buying include the demand-side platform, supply-side platform, and the ad exchange (Gonzalvez-Cabañas & Mochón, 2016).

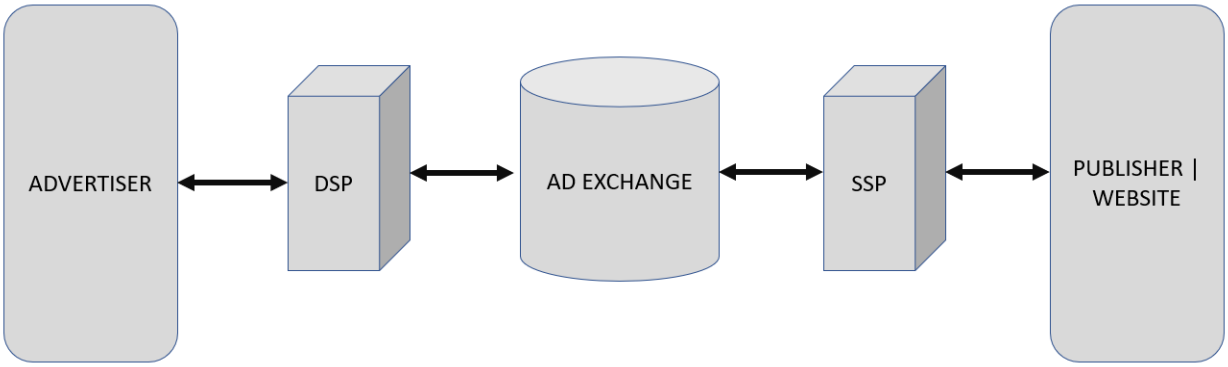


Figure 1. Illustration of the components of programmatic ad buying

The illustration in figure 1 depicts the major components that form a part of the programmatic ad buying technique. The functions of each of the components individually are given below to get a holistic picture of how programmatic ad buying works:

- Advertiser: The advertiser is the organization that is willing to purchase ad slots to serve their ads online. They have a creative team that builds out their ad creative and then approach a DSP (demand-side platform) to facilitate the purchase of ad slots on different websites programmatically over the internet.
- Publisher/Website: The publishers are the websites that have inventory available in their domain and are willing to put it up for auctions. Websites such as eBay or the guardian, for example, approach an SSP (supply-side platform) to facilitate the selling of ad slots available within their domain programmatically.
- DSP (demand-side platform): The DSP is a software that provides advertisers with the control over when and where they want to bid for ad slots that are available over the internet. The advertisers upload their ad creative to the DSP's and set up targeting strategies (different targeting strategies will be discussed further) to identify the ideal type of inventory to place a bid on.

- SSP (supply-side platform): The SSP performs a similar job for the publishers that the DSP does for the advertisers. Publishers approach the SSP to list their ad slots up for the auction.
- Ad Exchange: The ad exchange, as the name suggests, is where the auction takes place. The bid requests with specific targeting strategies set by the advertisers on the DSP are matched with the available inventory on the SSP, and the ad slots get auctioned. The advertiser with the highest bid for any ad slots wins the auction and gets to serve their ad on that inventory.

The entire bidding process within the ad exchange takes place within milliseconds and allows advertisers to bid for different ad slots across the spectrum of suppliers to serve their ads. This is essentially why programmatic media buying triumphs over the traditional media buying process and is seeing more traction in the space of digital advertising (Deshpande, 2019).

There are currently four main methods of buying programmatic media:

1. Real-Time Bidding (RTB):

   This is the most common method of purchasing ad inventory online programmatically. Websites that want to sell their ad slots put them up for auction. Advertisers bid for the ad slots in real-time and the most priced auction wins the ad slot and gets to serve their advertisement on it.

2. Private Marketplace (PMP):

   The private marketplace is very similar to the functioning of the real-time bidding process, in that the ad slots are auctioned between advertisers, and the one with the maximum bid gets to serve their advertisement. However, unlike the real-time bidding method, the websites put aside ad slots that they consider to be premium and only allow selected advertisers to be a part of the auction.

3. Preferred Deals:

   This is very similar to the traditional format of advertising. Here, the advertisers and the website get into negotiation and reach an agreement on a fixed price and defined durations during which the ads will be served on select inventory. This deal is done before the inventory is made available in the open exchange for accepting auctions from other advertisers.

4. Programmatic Guaranteed:

   This format, also known as programmatic direct, is very similar to the preferred deals or traditional media buying approach. Here, the advertisers and websites get into a one-on-one conversation to decide on the pricing, targeting strategies, duration, etc. of the advertisements, and the inventory on the website is sold directly to the advertiser. The difference between programmatic guaranteed technique and the preferred deals technique is that in preferred deals, the advertiser just gets a sneak peek into the type of inventory available, however, the ad-buying needs to still be done through a DSP. In programmatic guaranteed, the ad buying is done via separate contracts, and not through the DSP. Therefore, the advertiser is provided with complete information on what the audience of their ads would look like.

Having understood how programmatic ad buying works, let's dwell on the different targeting strategies available on the DSP's, which will give an insight into the problem statement and the gap that this study is trying to fill.

Unlike other forms of advertising, programmatic digital advertising gives advertisers greater flexibility on the type of audience they want to reach out to depending on the user demographics or the type of websites that they historically visit (Goldfarb & Tucker, 2011). This is because advertisers have at their disposal different types of inventories belonging to different verticals such as travel, gaming, etc. that can be leveraged to narrow down their ideal set of users. This flexibility feeds into the type of targeting that is available in digital advertising to serve ads on the internet.

There are essentially two main ways of ad targeting in the digital marketing platform:

- Contextual Targeting: involves the targeting of advertisements on specific websites or content that have been identified by the advertiser as relevant to their brand
- Audience Targeting: involves the targeting of advertisements to specific users, having analyzed their demographics by capturing their web cookie ids

Advertisers split their ad budgets across these two categories to reach and bid for the ideal ad inventory available on the internet. These targeting strategies are set up on the DSP by the advertiser before the bidding is done so that advertisers can bid on websites that they feel are relevant. The ad exchange filters out the inventory it receives from the SSP to only bid on those that match the targeting strategies set up by the advertiser before placing the bid.

2.4 Government Regulations on Web Cookie Data

Over the years, digital advertising has grown leaps and bounds, and with that, targeting strategies have evolved to become smarter to ensure advertisers can accurately reach the users that they believe are their ideal potential customers. However, with the advancements in targeting strategies, cookie data has become more prevalent and important in capturing trends of a user's online behaviour.

With advertisers now increasingly using cookies to collect data and storing them for campaign optimizations, governments have now started chipping in to regulate the collection and holding of cookie data as the data directly ties back to a user's online behaviour and could be seen by some as an invasion of one's privacy.

The European Union, for instance, introduced the General Data Protection Regulation (GDPR), which is the new data protection law applying to all the member nations of the European Union (Bussche & Voigt, 2017). The regulation took effect from May 25th, 2018 but allowed some organizations a 2-year period to align themselves with the regulation.

Below is a topline summary of the regulations that are mentioned in the data protection act of the European Union and pretty much the norm across other countries' data protection acts (Savić & Veinović, 2018). The regulation requires organizations that collect, store, or process personally identifiable information such as web cookie data to abide by the following 6 principles:

- The organization needs to be precise with the reasons why they're storing or processing the data. They need to be transparent and fair with the means of storing the data and ensure it's being processed lawfully.
- The organization should only be store and process the data for the purpose and duration stated originally.
- The organization should ensure minimal collection and store of the data pertaining to the originally stated purpose.
- The organization needs to ensure that the data stored is as relevant, accurate, and updated to their knowledge as possible. This also involves continuous erasing or rectification of the data to ensure its accuracy.
- The organization should ensure that the data is only stored for the time that its purpose is being fulfilled.
- The organization should take responsibility for the security and confidentiality of the personal data that they're storing or processing.

The above guidelines apply to the companies that deal with data in any form from any part of the European Union. However, most of the countries across the world that have begun imposing data regulations follow a very similar set of guidelines on the storing and processing of personally identifiable information of citizens.

With the governments taking strict actions and imposing hefty fines on organizations not abiding by the regulations, organizations need to ensure they're on top of them all the time. This implies that organizations in the digital advertising space need to move away from the audience (cookie) based targeting strategies. The research behind contextual targeting strategies becomes vital as advertisers begin searching for more sophisticated targeting strategies.

2.5 Existing Methodologies for Contextual Ad Targeting and Optimization

This study presents an exhaustive literature review done on the current space of contextual ad targeting and the existing techniques that marketers are implementing to optimize their campaigns. This review would help identify the gaps that exist in the current scenario of digital ad targeting, which would serve as the motivation for the methodology designed in this study.

In the current scenario, contextual (or inventory) targeting done on a campaign is based on historical campaign performance data or intuitively identifying the websites that may be relevant to the advertiser and serving ads on those domains (Berry, 2019). Optimizations on campaigns are done over time by adjusting the inventory based on campaign performance to shift spends to websites that are performing better (resulting in more engagement/clicks from the user interacting with the advertisement).

The most important aspect of optimizing a digital campaign is collecting the data first. Once advertisers have figured out a way to ingest ad campaign performance data, they can leverage the data to perform further optimizations. The ad campaign data typically consists of fields pertaining to every single advertisement that was served under the advertiser from the DSP. Data points such as the website where the ad was served, the timestamp of when the ad was served, how the user engaged with the advertisement (amount of time spent viewing the ad, whether the user clicked on the ad or not), the device type that the ad was shown on, etc. are collected within the feed.

The collection of these data points can be enabled using several techniques, both from the suppliers' end and the advertisers' end. The advertiser can place trackers on their ads that collect data points such as the website where the ad was served on, the position of the ad on the webpage, the cookie id of the user that saw the ad, the timestamp when the user saw the ad, etc. Along with this, the supplier/website and other 3rd party data providers can enrich the dataset further by providing additional data points e.g. the brand safety score of any website, the presence of potentially malicious activity on the website, etc.

Currently, advertisers leverage the ad campaign data mentioned above to optimize their digital ad campaigns (Marouchos, 2020). Based on the literature research around campaign optimizations, the existing methods of contextual campaign optimization identified are as summarized below:

- Inventory blacklists and whitelists: Advertisers can identify websites that have shown poor performance previously and blacklist those websites to avoid wastage of spends.
- Adjusting inventory spends: Advertisers use the data at hand to get a sense of what inventory is more suitable for them, and shift spends towards those inventories to enhance performance.
- Ad positions on a webpage: Advertisers use the data at hand to analyze whether serving ads on a specific position on the webpage results in better performance, for example, whether serving a wide skyscraper ad at the very top of a webpage results in greater user engagement compared to a banner ad.
- Device and system-based optimizations: Advertisers use the data to identify whether users using certain device types or operating systems are likely to engage more with the ad or not. For example, iOS users showing better ad engagement than android users.
- Brand Safety: Advertisers can leverage data from vendors who have audited websites based on how brand-safe they are. Aspects pertaining to website brand safety such as click bots, explicit content, illegal content, etc. are analyzed by these vendors and advertisers can use these data points to build on their website blacklists or whitelists.

The above methods of contextual ad campaign optimization have proven to be relatively effective over time, but there do exist plenty of limitations:

- The techniques mentioned above are reactive. They require ad campaigns to be activated and leverage data points from the data, as it arrives, to optimize the campaigns.
- Some of the techniques used currently are based on the intuition of an advertiser on what to target contextually and are not entirely data-driven.
- The learnings from just the ad data cannot be used across different verticals of advertisers; learnings from, say, a retail advertiser campaign may not be relevant for another advertiser in the travel domain.

2.6 Datasets for Contextual Ad Targeting and Optimization

The existing methodologies primarily leverage the log level data generated from the digital advertisements that have previously been run. The log level feeds for the ad campaign data consist of fields pertaining to ads that were served under the advertiser via the DSP. Data points such as the website where the ad was served, the timestamp of when the ad was served, how the user engaged with the advertisement (amount of time spent viewing the ad, whether the user clicked on the ad or not), the device type or system details of the device that the ad was shown on, etc. are collected in the feed.

However, the primary problem with using these datasets for generating optimization plans is that they are not robust and flexible. The data is very specific to a certain advertiser and hence cannot be used for optimizations across different verticals.

The approach taken in this research involves the usage of web traffic data (Meiss et al., 2008) which explores how a user moves from one website to another on the internet. This data is used to create clusters of websites that are similar to each other or share a common user base, and hence can be clubbed together for targeting or campaign optimization purposes. As a result, this dataset is advertiser/vertical agnostic, hence the optimization plan that would be produced from it will be flexible and can be extended across the entire digital advertising landscape.

2.7 Gaps in Existing Methodologies for Contextual Ad Targeting and Optimization

As mentioned in the previous section, the motivation behind this research is to identify sophisticated contextual ad targeting strategies to help mitigate the impact that the deprecation of cookie-based targeting will have on the performance of ad campaigns. As discussed in the literature review, there exists plenty of limitations with the current methods of performing contextual optimizations on digital campaigns.

Going back to the problem at hand, this study aims to identify how to optimally target an advertisement so that it reaches the right users. Advertisers need to understand which websites are similar to each other in terms of the audience that they share so that they can spend on relevant websites wisely. This study proposes using a data-driven analytical solution to the problem at hand to derive a contextual targeting strategy that can help optimize digital ad campaigns. The research aims to cluster websites that share a common audience group, using web traffic data, which will allow advertisers to target ad slots available on relevant websites together for serving ads.

This research aims to fill the gaps discussed above in the following manner:

- It is based on web traffic data hence isn't dependent on any sort of ad campaign data.
- Since it is based on an open-source dataset, it doesn't require the advertiser to run a non-optimized ad campaign just for the sake of collecting data points to perform the necessary optimizations. The ad campaign plan generated from this study would be data-driven from day one of the campaign.
- The output is derived using advanced statistical and data-driven concepts on the dataset, and not just based out of raw advertiser intuition.
- Since this research is based out of web traffic data, the output derived is advertiser agnostic. This means that the optimization plan can be extended to different advertisers or verticals of advertisers willing to optimize their ad campaigns contextually.
- Along with addressing the limitations above, the research can also be extended to other applications in the digital advertising landscape, which are explored later in this study.

2.8 Graph-based Data Representation

As mentioned in the previous section, we're dealing with web traffic data that includes data points pertaining to the path taken by users to move from one website to another. This form of data can be represented in many different ways depending on the use-case. The data could be used in its original JSON format, or even in a tabular format, to perform the modelling on. However, the objective at hand requires exploring relationships between websites in a more intuitive manner, and hence the data in a tabular format may not be the most suitable format.

On exploring the different techniques available for data representation, this study narrowed down on using graph-based representation based on reviewing the publications that depict the essence of such a representation of data and the advantages that it has over other data representation techniques (Hamilton et al., 2017). This study takes inspiration from the research done around methods and convenience of representing data in form of graphs for scientific computations (Hagberg et al., 2008). On the dataset that is being considered for this study, this format of data representation portrays the websites as the nodes of the graph and the edges would represent the share of users that are common to a pair of websites.

A study has previously been done on the web traffic data that has been sourced in this study to understand the different characteristics that are exhibited by users on the internet (Meiss et al., 2010). The publication attempts to model the behaviour of users on the internet in the form of graphs and compares the model with other Markovian models to assess the performance. This study takes inspiration from the benefits that were evident in representing a dataset as a graph structure as explained in the publication.

Graph-based representation of data has also been leveraged to assess the performance of the PageRank model on the web traffic data (Meiss et al., 2008). The paper presents the visualization of the web traffic data in form of graphs and validating the PageRank model on the web traffic data to statistically predict the future requests that users would make based on their historical browsing data.

The extensive review done on the above publications emphasizes the point that representing the data in a graphical format enables capturing the relationships that holistically exist between the different websites, rather than just looking at it as local connections, giving it an edge over the tabular format of data. A web graph structure, in the case of this study, would give a sense of how closely related any two websites are to each other, despite not having any direct connection with each other. This form of representation permits researchers to numerically quantify the relationships between websites, as discussed in the upcoming chapters.

2.9 Geometric Deep Learning Techniques for Generating Embeddings

Data modelling has evolved over the years, moving from the classical regression and clustering algorithms to more complex methods, such as neural networks, that tend to explain the data better. Neural networks are loosely based on the human brain and try to capture patterns that exist in the data in a numerical format that can be used to solve data-driven problems (Gron, 2017).

Neural networks have seen applications in a vast area of domains: classification problems, advanced clustering techniques, vector representation techniques, feature engineering, speech recognition, character recognition, etc. (Abiodun et al., 2018) Recently, the application of neural networks has broken into the field of representation learning and in this research, it is this application that is leveraged to generate the output.

In this study, the task of a deep learning model is to generate a vector representation of all the nodes of the website graph structure. To achieve this, several architectures can be utilized. Some of the most widely researched techniques that can be used to serve this purpose are:

- DeepWalk: This is a graph traversal technique that generates a series of random walks over different nodes on a graph (Perozzi et al., 2014). These walks are then served as an input to train a recurrent neural network to generate a matrix representation of the nodes in the graph structure.

- Node2Vec: This is an extension of the DeepWalk technique, as it also involves traversing over the graph structure to generate random walks which are used to train an RNN. The resulting model is known as a skip-gram model. The difference between Node2Vec and DeepWalk is that node2vec gives the researcher control over the way the random walks would be generated, which means that the user has control over whether they want to focus on the local relationships on a graph or global relationships.

- Structural Deep Network Embedding (SDNE): This technique utilizes a different approach to the above techniques. Instead of generating random walks, it uses two different metrics: first-order proximity and second-order proximity (Wang et al., 2016). Essentially, it aims to capture similarities between nodes that either share a common edge or share common adjacent nodes. This allows the SDNE's to capture non-linear relationships in any structure. However, SDNE's are relatively computationally more expensive than the above two methods as they use deep autoencoders to account for the above-mentioned metrics, unlike the walk-based methods that involve a simple neural network with just a single hidden layer.

- Large-scale Information Network Embedding (LINE): This technique is an improvement in the SDNE technique discussed above. It defines a single objective function that aims to preserve the first-order and second-order proximity (Tang et al., 2015). To optimize the stochastic gradient descent, this technique proposes a novel edge-sampling algorithm to generate the embeddings.

The choice of technique to be used for this study boils down to the type of result expected concerning the application, while also considering the limited computational resources available at hand. Considering the two proximity-based techniques i.e. SDNE and LINE have plenty of limitations when it comes to their application in this study. These are computationally more expensive compared to the walk-based techniques and don't give the user any control over choosing the local and global relationships present in the graph. As a result, the deep learning technique opted for in this study is the node2vec technique.

2.10 Node2Vec for Generating Embeddings

When it comes to representation learning, neural networks have heavily been used in the field of natural language processing to represent text data in numerical form. The word2vec representation learning (Mikolov et al., 2013a) forms the base of the idea behind representation learning as it uses artificial neural networks to represent words of a sentence or paragraph in a numerical format.

The word2vec model is essentially a 2-layer neural network that represents words in a sentence on a vector space i.e. in a numerical format. The representation of the vector space aims captures the intuitive similarities and differences that exist between the different words in such a manner the words that carry similar meanings will be a lot closer to each other on the vector space compared to other words. These vectors that represent the words are referred to as word embeddings, and these embeddings are produced by training a skip-gram model (Mikolov et al., 2013b).

The learnings from the word2vec model that was explained above were then extended to the graph ecosystem. The research that was done in this space resulted in proposing a technique known as node2vec (Grover & Leskovec, 2016) which is a framework for vector representation of nodes in a network.

The technique is an extension of the word2vec research and is used to represent node-based graphical data on a vector space having captured both the local and global trends that exist in the graphs. Like the word2vec model, the node2vec model will generate node embeddings by training a skip-gram model to represent the nodes of a graph on a vector space. The skip-gram model will aim to capture the proximities of the nodes in form of Euclidean distances between the points on a vector space in a manner such that the nodes that are close together on the graph are also close to each other on the vector space.

This study proposes to extend the application of node2vec research in the digital advertising space as part of this research.

2.11 Dimensionality Reduction

As part of this research, there were plenty of constraints concerning the computational power and the resources available at hand. Processing graph-based data over neural networks can be a very computationally expensive process. Also, working with high dimensional data can lead to problems of multi-collinearity and hence need to be processed accordingly.

This study explores dimensionality reduction techniques such as linear & non-linear PCA (Naik, 2018; Sarveniazi, 2014) and how they can assist with reducing multicollinearity in the data. The principal component analysis aims to reduce the number of variables in the dataset while still trying to preserve the variance that exists in the dataset. It's a technique used to extract features linearly in the form of principal components by trying to preserve as much variance as possible in the dataset. Another technique that has been researched on was the t-distributed stochastic neighbour embedding. t-SNE is a non-linear method that uses the probability of similarity between data points to represent them on a low dimensional space (Maaten & Hinton, 2008).

Table 1. Comparison between PCA and t-SNE

PCA	**t-SNE**
Used for dimensionality reduction	Used for dimensionality reduction
Linear Technique	Non-linear Technique
It is a deterministic algorithm that doesn't require tuning of hyperparameters	It is a non-deterministic algorithm and allows users to tune hyperparameters
Tends to not work well on relatively large dimensional datasets	Tends to work well on an extremely large dimension dataset
Computes principal components using a covariance matrix of existing data points to transform the data	Calculates the probability of similarity between data points on high and low dimensional spaces and attempts to minimize the difference between them.
Performs a one-time computation of the Eigen properties of the data to get the principal components	It's an iterative process that uses gradient descent to measure the difference between the probabilities of similarity by minimizing the Kullback-Leibler divergence.

The reason for using both these techniques in the research is to identify whether the website traffic data that we're using to identify clusters of websites would show better performance using a linear dimensionality reduction technique (PCA), or a non-linear one (t-SNE).

2.12 Clustering and Segmentation of Data Points

Clustering forms the final step of the methodology, and hence it was essential to perform an exhaustive literature review on the existing clustering techniques, and how they can be leveraged in this study. The main idea behind clustering is to identify data points that are similar to each other and can be clubbed to form a segment or cluster of data points (Vathy-Fogarassy & Abonyi, 2013). This is a necessary step in this study when modelling a segment of websites that are close to each other in the graph representation, which would be used for the optimization of digital ad campaigns. Based on the literature reviewed, different clustering techniques were identified and implemented in this study and the output of each was evaluated against each other.

The K-means clustering is the first method that this study proposes to use based on the publications researched on the algorithm. On the whole, this algorithm initializes centroids on a vector space that has data points and computes the Euclidean distance between the centroids & the data points iteratively to identify the points closest to each centroid on a vector space and segment them together (Kanungo et al., 2002). Further refinements on the algorithm have been done previously which enables us to initialize the centroids as further away from each other as possible, which has shown better results (Arthur & Vassilvitskii, 2007). The evolved technique is known as the K-means ++ technique.

Another clustering technique that is implemented in this research is hierarchical clustering. This technique, similar to K-means, is a linear technique and works on the principle of computing Euclidean distances between data points. The idea is to select any two data points as initial clusters and iteratively merge other clusters that are closest to each other until the required number of clusters are remaining (Murtagh & Contreras, 2011). This process is known as agglomerative hierarchical clustering, and it’s what is used as part of the methodology in this study. Contrast to that is the divisive hierarchical clustering process that initially considers the entire dataset as one cluster and splits it into different clusters based on distance metrics. Essentially, the reverse process of the agglomerative clustering.

The above methods for clustering are linear and may not be able to capture any non-linear patterns in the dataset. Therefore, background research was performed on the existing non-linear clustering techniques that can be leveraged in this research. The DBSCAN (density-based spatial clustering of applications with noise) is a well-studied clustering algorithm that uses the radial distance from every data point to cluster data points together (Ram et al., 2010). This research aims to evaluate the performance of the above clustering methods to identify the method most suited for the web-traffic data that is being used to optimize digital ad campaigns.

2.13 Related Research Publications

A significant portion of the review done on the literature for this study was around exploring the related research publications. Research presenting the foundation of digital advertising (Bala & Deepak Verma, 2018) has been very helpful as a part of the literature review to understand the background of the domain. The literature review also involved surveying the existing techniques used for the optimization of digital campaigns (Agarwal & Shukla, 2013), along with how these techniques are used to bolster customer loyalty (García et al., 2019).

This study proposes the use of the browsing behaviour of users, in terms of the websites that they visit. There has been plenty of research done in the space of website clustering and URL analysis. Publications around the clustering of URL's using networking techniques like formal concept analysis (Zhang et al., 2018) served as the inspiration behind deciding the methodology for this study. Other works around analysis concerning URL's in the concept of digital advertising involve creating classification systems that aim to identify malicious websites using lexical features (Blum et al., 2010) on smaller scales, using online learning algorithms on big data (Lin et al., 2013) or other deep learning techniques (Le et al., 2018).

The dataset used in this study has also been studied in related publications for analyzing the online behaviour of users. The clickstream data has been used to detect anomalies in visitation patterns and profiling of the users on the internet using graph-based techniques (Hofgesang & Kowalczyk, 2006). The same data has further been used to train a PageRank algorithm that extracts topological features and relationships that exist in the dataset (Meiss et al., 2008).

The literature review presented so far discusses several publications that have served as an inspiration for this study, as they have been fundamental in understanding the existing scenario of digital advertising and the gaps that exist in the domain. Along with that, publications around the technical concepts have been presented in this chapter that was paramount in narrowing down on the techniques to be used in the methodology.

2.14 Discussion

Based on the exhaustive literature review done as part of this study, it could be established that there is plenty of scope for research in the digital advertising domain. The study was able to identify gaps that exist in this domain, and this research attempts to fill those gaps using the proposed methodology presented in the following chapter.

As part of the literature review, the current landscape of digital advertising is presented, along with the research done in the contextual targeting techniques currently in place. Currently, the techniques are either based out of intuition or are very mildly data-driven. Also, these techniques are reactive i.e. they require ad campaign to be activated and leverage data points from the data, as it arrives, to optimize the campaigns. This study aims to fill this gap by proposing a contextual targeting strategy that is completely backed by data.

The current contextual targeting strategies, that are even mildly backed by data, have plenty of limitations. The learnings from those datasets are very specific to an advertiser, or a vertical, and hence can’t be extended across the landscape. The technique proposed in this study aims to capture the behaviour of users on the internet, and hence the output strategy is one that can be extended across any kind of verticals or advertisers.

The implications of stricter government regulations pertaining to the use of web cookie data cannot be overlooked, as they pose a serious concern to the current user targeting strategies for digital ad campaigns. As the use of cookies gets deprecated, the level of sophistication in targeting strategies would be impacted heavily, and this is another gap that this study aims to fill. The research done would help bolster the armoury of contextual targeting strategies that

marketers have at their disposal, which would be useful as the cookie-based targeting gets deprecated.

The literature review and the exploration of related works helped identify the gaps that exist in the domain, which served as an inspiration for setting the objectives and methodology of this study. Overall, this study would carry plenty of significance in the field of digital advertising and has the potential to be used as the foundation for potential further research.

2.15 Summary

The literature review done as part of this study is exhaustive and covers all the concepts necessary to understand the domain and achieve the objectives stated in the previous chapter. The techniques discussed in the literature review have undergone a high level of exploration and scrutiny before narrowing down on the optimal ones selected for the methodology.

The literature review and background research presented in this chapter covers the following broad aspects of this study: The knowledge of the digital advertising landscape, the gaps that exist in the landscape concerning the limitations in targeting and optimization strategies, the motivation behind this study, and the researched technical data science concepts that have been leveraged and implemented on a sourced dataset to achieve the objectives defined in the previous chapter.

The chapter begins with discussing the digital advertising landscape and the space of programmatic advertising, where the different methods of display ad targeting and optimization strategies are explained. The chapter then presents the gaps identified post the literature review in the digital advertising landscape. This background research lays down the foundation of the domain of the research. The chapter proceeds to discuss the research done on different datasets and the most ideal ones for this research. Post this, an exhaustive review is done on all the technical concepts relevant to the objective at hand and evaluates the most relevant ones for this study. Finally, the chapter presents the related work done in this domain of study, and the gaps that this study attempts to fill.

CHAPTER 3

METHODOLOGY

The methodology that has been followed in this study is researched thoroughly and succinctly to meet the aim and objectives of the research mentioned in chapter 1. In this section of the thesis, the procedure followed to achieve the objectives is outlined. The section also describes all the data science techniques used in the procedure in detail, from their algorithm to the implementation. Overall, the methodology used as part of this research largely follows the widely recognized CRISP-DM framework and leverages the different components that form the core of the framework.

3.1 Introduction

To achieve the aim and objectives that this study initially set out to attain, there was a comprehensive course of action followed. From sourcing the dataset, to importing & pre-processing the data, to identifying and implementing the advanced statistical techniques on the dataset, to finally establishing a customized evaluation parameter to assess the outcomes, there was an exhaustive methodology that was followed as part of this study, which is presented in this chapter.

The chapter begins by discussing the technique used to import the dataset, the format of the dataset imported, and the pre-processing steps that were followed to transform the data into the appropriate format. This covers the data sourcing and pre-processing aspect of the research.

Post this, the procedure for converting the data into graph structure is outlined. The websites and their connections are represented on a graph that will be used to generate embeddings. There is deliberation done on the node2vec algorithm implemented in the research. A separate section covers the technical aspects pertaining to how the node2vec algorithm generates embeddings.

The chapter proceeds to explain the methodology used behind segmenting the websites once the embeddings have been generated. The various dimensionality reduction and clustering techniques implemented in this study based on the literature review presented previously have been discussed and presented in the sections that follow. Each of the different techniques has been tuned using various hyperparameters, depending on the processing resources that were available at hand, and the various versions of the models have been discussed.

Finally, the section closes with the discussion on the evaluation parameters that have been used to assess the different variants of the models created. To assess the final output, a customized evaluation parameter was designed to evaluate the different models and to identify the best performing model pertaining to the dataset at hand.

3.2 Data Sourcing and Preparation

In this section of the methodology, the methods used to source the data and prepare it for modelling are outlined. The section initially begins by describing how and from where the data was identified and imported. Post that, the section outlines the original structure of the data, the extensive pre-processing done on the dataset, and the final structure of the data that would form the foundation of the research that follows.

3.2.1 Data Selection

The very first step of the methodology involved sourcing out and identifying a recent yet relevant open-source dataset that would be used as the source dataset for this research. The dataset would have to have all the relevant information that would be needed to achieve the initial aim and objectives that were set out. Post performing exhaustive research on the various datasets and research papers available in the field of digital advertising, this study narrows down to using the dataset pertaining to the web traffic requests for November 2009 (Meiss et al., 2010). The dataset is apt for this research as it constitutes of all the fields that are relevant to the objective devised.

The dataset being used in this study is pertaining to data collected from edge servers of the Indiana University network. The servers collect the HTTP requests made on the network and report the data back to the holders that maintain the record in raw data files. This is by far the most exhaustive dataset pertaining to user HTTP requests based on the secondary research done and can be assumed to represent the browsing behaviour that would be followed across the network. The study ahead can be used as a proof of concept for every potential dataset pertaining to digital advertising that collects cookie-based data from users based on their browsing behaviour.

The dataset sourced from the internet is aggregated at an hourly level and contains four columns:

1. Timestamp: this is the hour at which the data is aggregated.
2. Referrer: this is the referrer website; the column contains the website from where a user originates.
3. Destination: this is the target website; the column contains the website where the user goes to from the referrer website.
4. Count: the number of users in the hour that follow the path from the referrer to the target website.

The data is downloaded from the internet and loaded into the python environment where the entire methodology ahead is executed.

3.2.2 Data Pre-processing

The data sourced from the internet, as mentioned in the previous section, required plenty of pre-processing before it would be ready for modelling purposes. This sub-section discusses the various data preparation techniques that were required to be implemented on the dataset to have it ready for further steps.

The data sourced from the website was initially in a compressed tar archive file that required uncompressing to be accessed. Post uncompressing the data, the uncompressed data files were not available in the customary tabular format. The files were in the JSON format and required further processing to convert to a processable tabular format. The dataset was parsed in python

and cleaned up accordingly so that the required columns, mentioned in the previous section, are loaded in a structured, tabular format in the form of a python data frame.

Once the data is in the tabular format, a sense check needs to be performed on the different columns. Beginning with the data types for each column, the necessary data types are assigned explicitly for each one of them i.e. the website columns as a string, the timestamp (which is currently the Unix-timestamp), and the user count columns as integers.

In the case of the two website columns, there needs to be a lot of textual data cleaning to be done. Below mentioned is a list of data cleaning steps done for each of the website columns along with the necessary filtering done to obtain the final refined dataset:

- Some of the websites provided in the dataset are entire URLs of web pages i.e. they include the domain name along with the remaining section of the webpage. Therefore, all sections of the URL are eliminated apart from just the website domain name. This is done by removing all character after (and including) the character '/'.
- The data consisted of websites from across different countries and organizations. To have an intuitive and concise set of websites, only the domains with extensions relevant to the UK were considered i.e. domain.com, domain.net, domain.co.uk, domain.edu, domain.org, domain.gov
- Filtering out websites that contain punctuation characters that are not associated with the usual format of the websites e.g. '%', '/', '\', '"', ';', '='.
- Filtering out websites that hold explicit/unrestrained content.
- Filtering out all the combination of websites that have less than 50 users in common.
- Filtering out entries from the dataset that have the same referrer and destination websites.

Having performed the cleaning on the websites, the data on the number of users that have moved from one website to the other was aggregated across the entire month, rather than the hourly level that was originally mentioned. As a result, the timestamp column was dropped, and the dataset was aggregated on the count of users for the different website combinations. Eventually, the dataset was left with three columns i.e. the referrer website, destination website, and the total number of users that followed that path across the month.

3.2.3 Graphical Representation of Data

A graph is essentially a collection of nodes on a space interconnected with edges or links that depict the relationship that exists between the nodes. In a connected graph, the nodes that are similar to each other would be relatively closer to each other compared to other nodes i.e. the length of the edge would be smaller. Such a network of nodes can be used to extract relationships from the data that would otherwise not be intuitive from a dataset in a tabular format. Chapter 2 introduced the motivation behind implementing the graph structure to the dataset as it helps in capturing the local and global relationships between datapoints located on the graph. To convert the dataset into a graphical structure, the functions available in the networkX library in python is leveraged (Hagberg et al., 2008).

The networkX library has in-built functions that accept parameters such as the source node column, destination node column, and the attribute corresponding to the edges. The function converts the tabular dataset into a graphical structure, where the nodes of the graph correspond to the websites, and the edges connecting the graph correspond to the links between the websites. This is a weighted graph, and the weight (length) of the edges is inversely proportional to the number of users shared by any two websites i.e. the websites that share a larger user count are closer to each other compared to other websites.

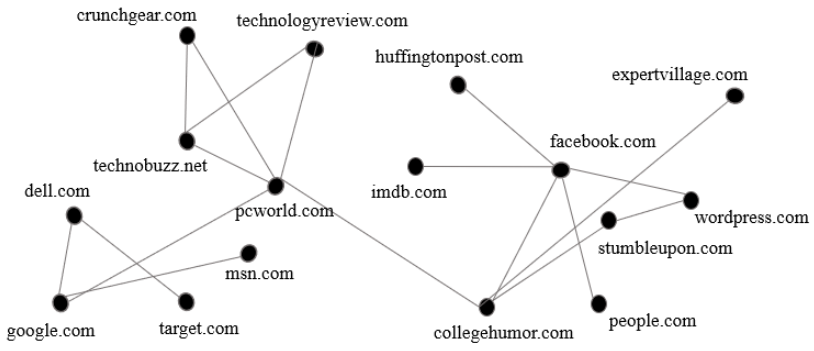


Figure 2. Graph-based representation of websites

The image in figure 2 represents a small section of the entire graph that is produced off the back of the original dataset. The graph is used for modelling purposes in the upcoming sub-sections.

3.3 Generating Embeddings from the Graph Structure

In this sub-section of the methodology, the applied geometric deep learning techniques on graph structures are discussed. Initially, there is an overview of embeddings and how neural networks can be used to generate them. This proceeds with an explanation of the working of the model that has been used as part of the research, and finally, the implementation of the model on the graph structure designed in the previous section is presented.

3.3.1 Geometric Deep Learning Based Embeddings

Embeddings, in plain terms, refers to the representation of data on an n-dimensional vector space. This could be anything such as words of a sentence, sentences in a novel, users on a social network, employees in an organization, etc. Embeddings capture the relationships that exist between data points and represent them on a vector space allowing for mathematical computations on the data. They have widely been researched in the natural language processing space, where deep learning techniques have been used to capture the relationships between textual objects to model them. This is what led to the word2vec algorithm (Mikolov et al., 2013a), which was used to identify relationships between different words and represent them on a vector space.

The learnings from the word2vec algorithm were then later extended to the graph structure which led to the node2vec algorithm. The node2vec algorithm was created to represent the nodes existing in a graph structure on a vector space by capturing the relationships that exist between different nodes. The node2vec algorithm was later extended to the graph2vec algorithm, which is essentially trying to represent entire graphs, or subgraphs, on a vector space to model the relationships between them (Hamilton et al., 2017).

As part of this study, the node2vec algorithm has been implemented on the graph structure that was prepared as discussed in the previous sub-section. In the next sub-section, the node2vec algorithm is crisply explained to develop an understanding of how it’s implementation results in the embeddings of the websites.

3.3.2 Node2Vec

As discussed in the previous section, the node2vec algorithm uses a neural network to represent the nodes of the graph structure on a vector space. This sub-section dwells into the functioning of the entire model, and how the nodes are converted to a vector representation with the help of the model.

There are essentially two major components that pertain to the node2vec algorithm:

1. Generating a 'corpus' from the graph structure.
2. Training a skip-gram model on the 'corpus' to generate embeddings.

The 'corpus' mentioned above is a set of connected nodes that can be used to train a model on. The analogy to word2vec here would be how the words are connected in a sentence like the nodes being connected in a graph.

The method used to generate the corpus (sample connections of nodes in a graph) is by taking random walks from every node in the graph (Sajjad et al., 2019; Li et al., 2015), based on the transition probabilities that exist for moving from one node to the other in the graph. Therefore, for every node on the graph, a random walk is generated by moving to adjacent nodes connected to that node in an order governed by the transition probabilities between the nodes.

To denote the transition probability for a random walk between the current node (c_i), *x* and the previous node (c_{i-1}), *v*:

$$P(c_i = x \mid c_{i-1} = v) = \begin{cases} \frac{\pi_{vx}}{Z}, & \text{if } (v, x) \in E \\ 0, & \text{otherwise} \end{cases}$$

Where π_{vx} represents the transition probability and is normalized by a constant Z, between any two nodes *v* and *x*.

Along with the transition probabilities that are discussed above, the random walks in node2vec are also governed by the weights of the edges between two nodes. To get a better intuition of how the random walk concept is implemented, consider the illustration of a graph shown in figure 3. A random walk from node A of the graph with four steps could result in a path as shown in the illustration. The transition between nodes is governed by the transition probabilities and the edge weights between the nodes. Similarly, random walks are made from all the nodes of the graph to generate corresponding paths that form the entire corpus of the graph structure, which would be fed to a skip-gram model for generating embeddings.

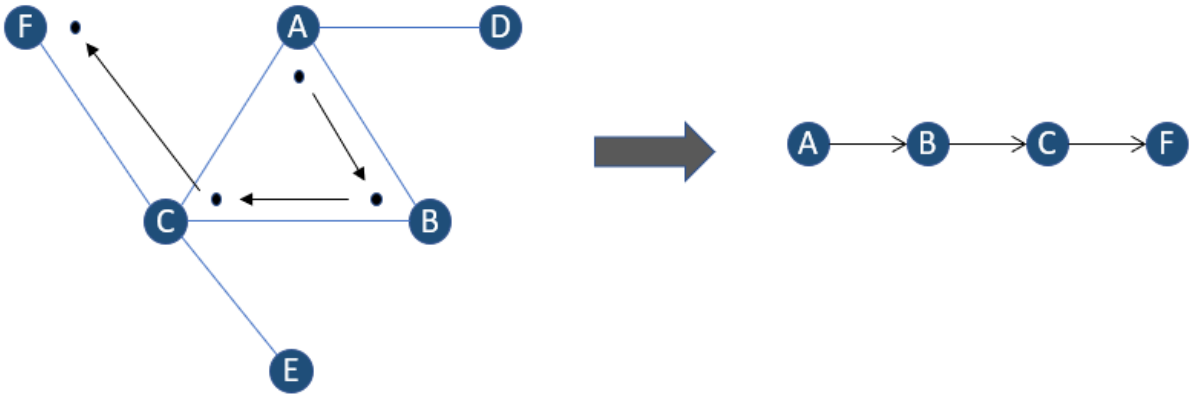


Figure 3. Random walk generation

The random order, however, is also governed by the hyperparameters that are selected in the initiation of the walk. Before delving into how these hyperparameters work and how they govern the corpus that is created out of the random walks, let's understand the different search strategies that can be implemented on a graph using random walks.

There are essentially two sampling strategies that can be adapted when generating the random walks to model the relationships that exist between nodes in a graph:

1. Breadth-first Sampling (BFS): Here, the emphasis is given to the nodes that are the closest neighbours to the initial node. As a result, the random walk will give a priority and move to the nodes that have the smallest distance to the initial node.

2. Depth-first Sampling (DFS): Here, the random walk attempts to capture the global relationships that may exist in a dataset, and hence the priority is given to nodes that are placed further away from the initial node while generating the random walk.

Both these random walk sampling techniques present the extreme case scenarios, and the hyperparameters in the node2vec technique are used to govern the sampling of the random walk based on how one opts to model the relationships. The BFS sampling technique would result in representing nodes that are closer to each other on the graph as having smaller distances between each other on a vector space. Whereas, the DFS sampling technique would result in representing the nodes that are further away from each other on the graph as having smaller distances between each other on a vector space.

On identifying the most relevant set of hyperparameters based on the objective that is set to achieve, random walks are generated which results in creating the corpus, which is essentially a set of paths of nodes across the graph. This corpus is then used to train a single hidden layer neural network, which is referred to as a skip-gram model.

A skip-gram model is a basic model created by training a single hidden layer neural network on a nodal classification task in the case of node2vec. The classification task is not the primary objective of the skip-gram model, but the weights of the neural network that are trained during this task are what is of the essence in this case. These weight vectors of the hidden layer are referred to as the nodal embeddings, or simply the representation of the nodes on a vector space.

Neural networks expect the input to be served in a numerical format, and therefore the nodes need to be converted into numerals. As a result, in the simplest form, all the nodes in the graph are one-hot encoded and fed into the neural network to train a skip-gram model.

The neural network will be trained to achieve the following: for any chosen node in the random walk fed into the model, the network will predict the probability of all other nodes being the adjacent node to the chosen node. Therefore, the output probabilities for every node in the skip-gram model correspond to how likely that node is to be adjacent to the chosen node.

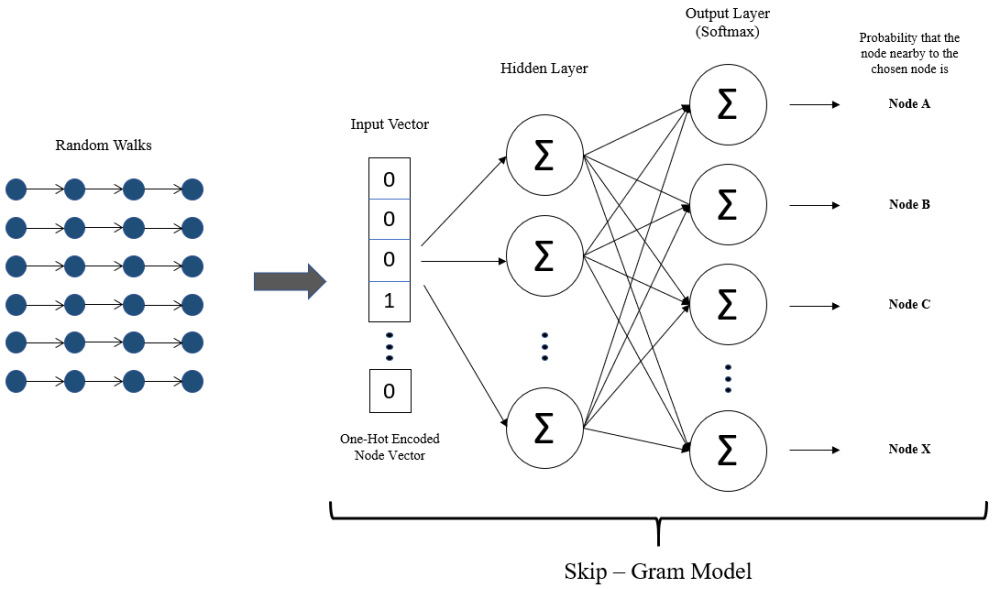


Figure 4. Node2vec skip-gram model

Effectively, the skip-gram model is trying to optimize the embedding function, f(u), that maximizes the log probability of any node, u, finding its neighbouring nodes, $N_s(u)$ on the graph. This optimization function of the skip-gram model can be expressed as follows:

$$\max_f \sum_{u \in V} \log P_r(N_s(u)|f(u))$$

The above-mentioned explanation is the dummy task that the skip-gram model is trained to optimize. However, rather than the output layer of the skip-gram model, it's the hidden layer is of importance to the user. The hidden layer has no activation function, whereas the output layer is a SoftMax function. This means that the hidden layer would essentially just be the weight matrix which is made up of vectors corresponding to every different node.

Within the above explanation lies the intuition behind the node embeddings that are generated. The training of the neural network would result in nodes, that are close to each other or connected, being represented by vectors that are very similar to each other in the weight matrix of the hidden layer. These hidden layer weight vectors are the actual output of the node2vec algorithm and are referred to as the node embeddings.

3.3.3 Node2Vec Implementation

The previous sub-section provides a holistic understanding of how the node2vec algorithm works, and the intuition behind how the algorithm can be used to generate website embeddings as part of this research. This sub-section provides the implementation of node2vec in this research and the hyperparameters are chosen to achieve the results.

The implementation of the algorithm above has been done in this research using two python libraries: node2vec and genism. These libraries have functions that enable the training of the skip-gram model on a graph structure, along with giving the user the flexibility to tune the model.

From the last touchpoint of the methodology section, the dataset was converted to a graph structure with the graph nodes as websites and the graph edges as the count of users between the websites. This graph is fed to the node2vec algorithm to generate the node embeddings. The algorithm follows the same procedure discussed in the previous section, whereby, random walks are generated which are used to train a skip-gram model, and the resulting weight vectors of the hidden layer are effectively the node embeddings for individual nodes.

However, the part of generating the random walk has been monitored carefully using hyperparameters depending on the objective that is being attempted to achieve in this research. As discussed in the previous chapter, the random walks can be generated using two techniques, BFS and DFS, depending on the relationship that is needed to be captured in the graph. As part of this study, the embeddings that need to be generated for nodes are supposed to be representative of the number of the weight of the edges between the nodes i.e. the number of users common to the websites. Hence, the BFS technique is the ideal method that needs to be considered for generating the random walks, as it prioritizes the local relationships between nodes in a graph, and hence would be able to capture the nodes that have larger weights between them as part of the random walks generated. This would eventually result in the generation of embeddings that depict the true relationship that exists between websites with regards to the number of users that are common to them.

To take the above into account, the node2vec implementation would require tuning of hyperparameters. To emphasize the BFS technique within node2vec, two specific hyperparameters need to be controlled:

- The return parameter, p, that controls the likelihood of the random walk traversing over the same node again. A high p value encourages the random walk to explore more nodes rather than traversing over the same node again. As a result, in this research, this parameter is set to > 1.
- The in-out parameter, q, that controls the exploration aspect of the node traversal during a walk. A small value of q would result in a random walk attempting to capture nodes that are further away from each other, essentially capturing the DFS aspect of the graph. As a result, in this research, this parameter is set to > 1.

Along with this, other crucial hyperparameters have been set in the implementation of node2vec in this research:

- The number of walks – corresponds to the total random walks that need to be derived for every node. This has been set to 100, which is a value large enough to capture the relationships that exist around the nodes in the web graph, considering the total number of websites in the graph.
- Walk length – governs the number of nodes that will be traversed in every random walk. This has been set to 10, which is a value that along with the number of walks would help capture the local relationships existing between the websites well.
- Dimension – governs the length of the weight vectors that would be used when training the skip-gram model. The embeddings are evaluated using three different sets of dimensions as hyperparameters: 10, 20, and 50.

Post implementing the node2vec algorithm on the graph dataset with the tuned hyperparameters, the website embeddings are generated, which is essentially the vector representation of the websites on a vector space. In the following sub-sections, these embeddings are validated and evaluated to check the quality of the procedure.

3.4 Dimensionality Reduction

In the previous sub-section, the node2vec algorithm was discussed along with its implementation that resulted in the generation of website embeddings for three different sets of dimensions as hyperparameters. Post generation of the embeddings, dimensionality reduction techniques are implemented to check whether the different dimensions exhibit any multicollinearity and if the variance that exists in the data can be captured in a fewer set of dimensions.

In this sub-section, two different techniques of dimensionality reduction are explored and implemented on the dataset. The choice of the technique to be used for this study was decided based on the literature review done on dimensionality reduction as presented in Chapter 2. The evaluation would be done on both these techniques to compare the more effective one of the two, later in this study.

3.4.1 PCA Implementation

The first technique on dimensionality reduction implemented on the embeddings generated is the principal component analysis. This is a linear technique that computes principal components for the existing vector representation using a covariance matrix to transform the data.

The algorithm computes eigenvalues and eigenvectors for identifying the principal components. The principal components, in this case, are the new features that have been formed, and a combination of the different principal components would explain the variance in the dataset in a finer manner.

The computation of the components would be in such a way that the first principal component would explain the highest amount of variance, followed by the second, and the rest would follow a descending order of variance being explained. The total number of principal components would be equal to the total number of dimensions in the dataset (Mishra et al., 2017).

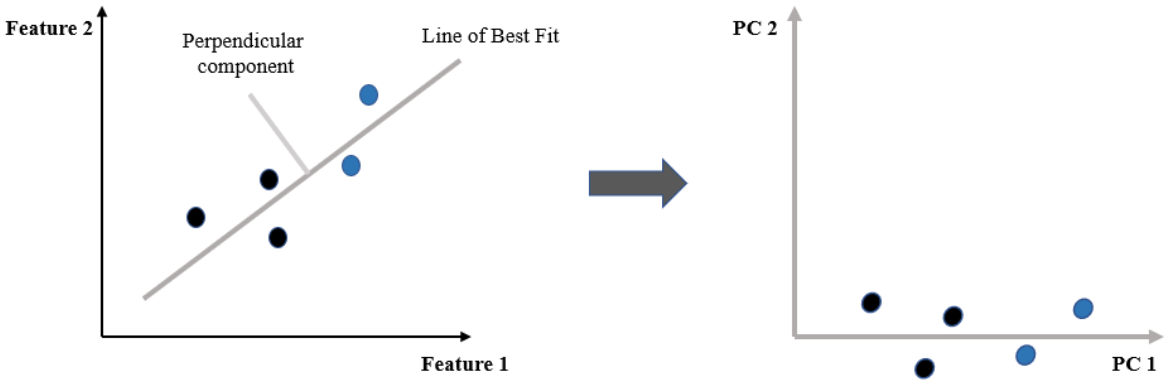


Figure 5. Illustration of dimensionality reduction using PCA

The illustration shown in figure 5 gives an intuition on how the principal components are created. Consider the 2-dimensional vector space on the left; to compute the principal components, a line of best fit is plotted across the points. This line effectively forms the new axis and can be considered as the 1st principal component of the data, explaining the highest amount of variance in the data.

The 2nd principal component would be an axis that is perpendicular to the 1st principal component and would explain minimal variance. This is where the concept of dimensionality reduction applies, whereby the 2nd principal component can be omitted from the data, and the majority of the variance would still be captured by the first principal component.

Post generating the embeddings, as discussed in the previous sub-section, the PCA technique explained above is implemented on the embeddings using the PCA function in the sklearn python library. As a result, any multicollinearity that would exist between the dimensions of the embeddings generated would be identified and gotten rid of. The output of this procedure would be a refined set of embeddings, effectively the principal components, that can be clustered to create website groupings, which is the aim of this study. The evaluation of this technique is done and discussed later in this methodology section.

3.4.2 t-SNE Implementation

t-SNE, or t-Distributed Stochastic Neighbor Embedding, is another technique that is used for dimensionality reduction. Unlike the PCA, this technique doesn't linearly reduce the dimension and hence is one of the ideal techniques to capture non-linearity that exists in the data when reducing the dimensions. Unlike the PCA that computes linear principal components, the t-SNE technique creates a probability distribution to map the points onto a lower-dimensional scale. This probability distribution follows a t-distribution, hence giving it the name t-SNE.

The below points give an intuition on how the algorithm works:

- For every datapoint x_i, the local relationships are captured by computing a probability distribution for every other point x_j based on how close the point is to the base point, for reasonable values of the standard deviation σ_i i.e. the points that are closer to the reference point would have a higher probability in the distribution.

$$p_{ij} = \frac{\exp\left(\frac{-\left|\left|x_i - x_j\right|\right|^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(\frac{-\left|\left|x_i - x_k\right|\right|^2}{2\sigma_i^2}\right)}$$

  The value of σ_i is chosen in such a way that it's low for the points that are closely placed and high for the points that are sparsely placed. A hyperparameter known as perplexity is tuned to achieve this, which is essentially specifying the number of stochastic neighbours that should be considered for each datapoint.

- The algorithm then attempts to map this probability distribution on a lower-dimensional space. Say, y_i is the representation of every data point x_i on the lower-dimensional space, and the variance of the distribution is set as $\frac{1}{\sqrt{2}}$, the distribution would be expressed as:

$$q_{ij} = \frac{\exp\left(-\left|\left|y_i - y_j\right|\right|^2\right)}{\sum_{k \neq i} \exp\left(-\left|\left|y_i - y_k\right|\right|^2\right)}$$

- The function that is to be optimized is the Kullback-Leibler (KL) divergence between the above two distributions p and q. A gradient descent technique is implemented to optimize the KL-divergence as follows:

$$\frac{\delta J}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)\left(1 + \left|\left|y_i - y_j\right|\right|\right)^{-1}$$

The t-SNE technique has been used to perform a non-linear dimensionality reduction on the embeddings generated post the node2vec technique. The reason for performing both these techniques is to evaluate the quality of the embeddings using both the linear and non-linear dimensionality reduction techniques. The output of this procedure would effectively be the lower-dimensional mapping of each of the embeddings based on the optimized probability distribution, that can be clustered to create website groupings, which is the aim of this study.

3.5 Clustering the websites

The previous section delved into the techniques used for reducing the dimensions of the embeddings. The reduction of dimensions is necessary to avoid the existing multicollinearity in the data. Post this, the study delves into the third objective function i.e. clustering the websites using the embeddings' dataset to identify groups of similar websites that can be used for contextual targeting of digital ad campaigns. In this chapter of the methodology, there are three different clustering techniques explained that have been implemented in the study. The reason for implementing all three of them is because each of the algorithms uses a different approach for clustering the websites, and this study will later evaluate all the algorithms to identify which one was best able to express the relationships that exist between the websites in form of the clusters.

3.5.1 K-Means Clustering of Websites

The first technique explored in the clustering of websites is the k-means algorithm. This is a linear clustering algorithm that attempts to identify the linear relationships between the data points and groups the data points based on the Euclidean distances. The basic objective of the k-means algorithm is to minimize the variance that exists between data points within a cluster and maximize the variance between data points of different clusters.

Before beginning with the application of the k-means algorithm, there are a few checks that need to be done on the dataset to identify whether it is ready for the clustering process. Before clustering the data, it needs to be assessed for the clustering tendency (Adolfsson et al., 2019). The study uses the Hopkin's statistic to evaluate the clustering tendency of the data. The Hopkin's statistic measures the probability of how uniformly distributed a given dataset is, and scores the data from 0 to 1, with a score of 1 implying the dataset has a very high clustering tendency.

Let x_i represent the distance between every data point to their nearest neighbour on the dataset, and y_i represent the distance between a randomly distributed data points on the dataset and the closest actual data point on the dataset, the Hopkin's statistic is calculated as:

$$H = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i}$$

Post evaluating the clustering tendency, the clustering technique is implemented. However, the k-means clustering algorithm has a vital initial requirement: the number of clusters to be formed needs to be initialized. To identify the optimum number of clusters, the silhouette analysis is implemented. The silhouette analysis is performed post the clusters have been generated and indicates how well the clusters have been formed i.e. whether the variance of data points within the same cluster is low and those outside the cluster is high (Wang et al., 2017).

The silhouette value or coefficient for any clustered dataset is calculated as:

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}$$

Where a_i represents the mean distance between every data point to all the data points outside the cluster, and b_i represents the mean distance between every data point to all the data points within the same cluster. This is computed for all the data points in the dataset, and the mean value of all the s_i values in the dataset give the average silhouette coefficient for the dataset s_k

To identify the optimal number of clusters, the k-means algorithm was iterated over a range of an initial number of clusters, from 5 to 50. For each iteration, the silhouette value would be computed and plotted. The number of clusters chosen would eventually be decided as the one that resulted in the highest value of the silhouette coefficient, SC.

$$SC = \max_k s_k$$

Post the above analysis, the k-means algorithm is implemented on the embeddings' dataset using the functions pertaining to this algorithm in the sklearn python library. The following procedure gives an intuitive understanding of how the k-means algorithm generates the clusters of websites on the dataset:

- The optimal number of clusters, k, from the silhouette analysis is initialized.
- The k-means ++ technique is leveraged to initialize the centroid positions. This technique assigns k number of data points in the dataset as the cluster centroids that are furthest apart from each other.
- Once the initial centroids have been selected, the first iteration of k-means computes the Euclidean distances of every data point to the cluster centroid and assigns the data point closest to it to that cluster.
- Once the assignment step is completed, the new cluster centroid is computed.
- The steps 3 and 4 are iterated until all the data points are assigned to the k clusters and no further changes happen.

The output of the k-means algorithm is a set of website clusters that are close to each other on the embeddings' vector space. The websites within each segment are similar as they share a common user base, and therefore can be targeted together for contextual optimization of digital ad campaigns.

3.5.2 Ward's Hierarchical Clustering of Websites

The next form of clustering explored in this study is the agglomerative hierarchical clustering technique. Like the k-means, this technique is also a linear technique, that computes clusters based on the Euclidean distances between data points. However, the algorithm follows a different approach to the k-means algorithm.

To implement hierarchical clustering on the embeddings, the functions available in the sklearn python library pertaining to this algorithm were leveraged. Clustering happens as follows:

- For the dataset with N number of data points, a distance matrix is computed that holds the Euclidean distances between every data point to all other data points.
- Initially, all the data points are assigned as individual clusters, hence there are N clusters.
- The closest pair of clusters are identified and merged. Therefore, at the end of this step, there are N-1 clusters.
- The distances between the new cluster and all other data points are computed and a new distance matrix is created.
- Steps 3 and 4 are repeated until all the clusters have merged into one cluster of size N.

Post iterating through the above algorithm, a graph known as a dendrogram is computed that depicts the above procedure, and how the clusters were generated. The illustration shown in figure 6 presents a sample dendrogram graph created by performing agglomerative hierarchical clustering on 8 data points.

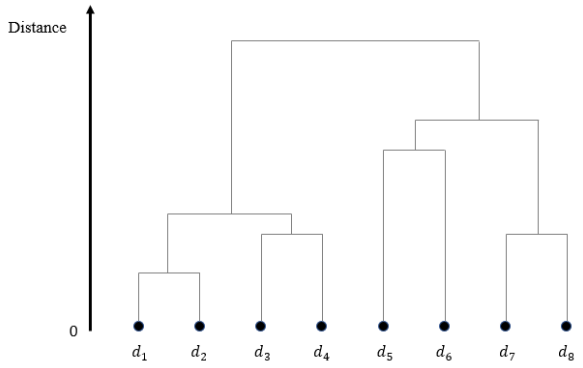


Figure 6. Dendrogram for agglomerative hierarchical clustering

The most important aspect of the above algorithm is the distance computation that happens between clusters. This is referred to as the linkage that exists between the clusters, and in this study, Ward's method of computing the distance between clusters at every iteration (Murtagh & Legendre, 2011) is used.

Ward's method implies that the distance between any two clusters A and B is expressed as:

$$\Delta(A,B) = \frac{n_A n_B}{n_A + n_B} ||m_A - m_B||^2$$

Where n_i represents the number of points in a cluster and m_i represents the centre of a cluster. Intuitively, Ward's method attempts to optimize the above objective function, which aims to minimize the intra-cluster variance and maximize the inter-cluster variance, to identify the clusters to merge at every iteration. The optimal number of clusters to be eventually produced is based on the similar approach discussed in the k-means algorithm. The silhouette analysis is applied here as well to determine the optimal number of clusters.

The output of the hierarchical algorithm is a set of website clusters that are close to each other on the embeddings' vector space. The websites within each segment are similar as they share a common user base, and therefore can be targeted together for contextual optimization of digital ad campaigns.

3.5.3 DBSCAN Clustering of Websites

The above clustering methods may not be able to capture any non-linear relationships that exist between the data points in the dataset. The DBSCAN clustering algorithm is a density-based clustering algorithm that is non-linear and can capture any spatial relationships that may exist in the dataset. Essentially, the algorithm attempts to cluster regions in the dataset that are highly dense and separates them from regions that are of low density.

In this study, the functions in the sklearn python library pertaining to this algorithm have been leveraged to implement the clustering. The algorithm requires tuning of two major parameters:

- Eps, ε - represents the radial distance around every datapoint that the algorithm will look out for in search of data points to form a cluster.
- Min Samples –represents the minimum number of data points needed to form a cluster.

The DBSCAN algorithm assigns every data point into three different groups:

- Core points – these are points that are successfully clustered by the algorithm.
- Border points – these are points that are reachable from the core point, ε, but there are less than the minimum number of data points required around it.
- Outlier points – these are points that were not clustered and termed as anomalous.

Figure 7 illustrates the typical output from the DBSCAN algorithm, and how the three different types of points look like with respect to the clusters created.

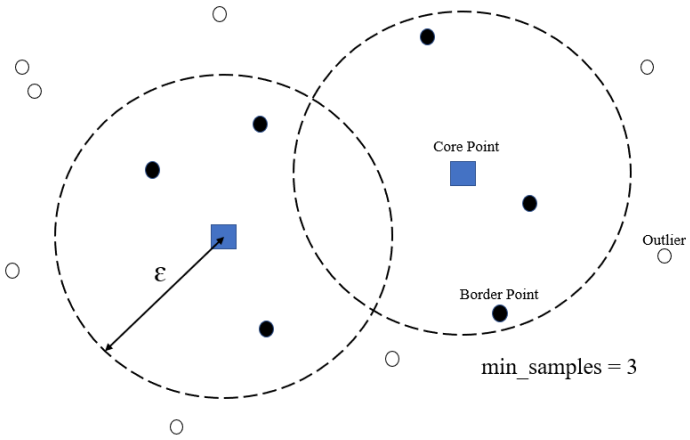


Figure 7. Clustering output from the DBSCAN algorithm

The output of the DBSCAN algorithm is a set of website clusters that are close to each other on the embeddings' vector space. The websites within each segment are similar as they share a common user base, and therefore can be targeted together for contextual optimization of digital ad campaigns.

3.6 Evaluation/Interpretation

So far, three out of the four objectives have been achieved. The final objective involves evaluating the clusters created from different hyperparameters and techniques. A custom evaluation criterion is designed to score these clusters. Every cluster would be judged based on how well the websites it holds are connected.

The first metric that is designed aims at capturing the dissimilarity that exists between websites in a cluster. This is referred to as the cluster error, CE, and refers to the share of websites in a cluster that has no links with any other website in that cluster, and is expressed as:

$$\text{Cluster Error, CE} = \frac{\text{Number of websites in a cluster that don't share common users with any other website in that cluster}}{\text{Total number of websites in the cluster}}$$

The clusters are scored using the Website Overlap score as follows:

$$\text{Website Overlap, WO} = 1 - \text{CE}$$

The website overlap score, ranging between 0 to 1, indicates the share of websites inside a cluster that have users common to each other. A high website overlap score indicates that the cluster is made up of websites that are closely related to each other, which is ideal.

In addition to the website overlap score, another parameter that needs consideration is the weight of the clusters. In the context of digital advertising, it would be ideal to have an equal number of websites in each cluster when generating segments for targeting. Consider a case where out of say 15 clusters, one cluster occupies 95% of all the websites and the rest of the 5% of websites

are distributed across the different clusters. Despite the website overlap score being high here, it is not a desirable output. To take this into account, a parameter known as the cluster skew is introduced that penalizes any cluster for having clusters more than or less than the average number of websites expected in the cluster.

$$\text{Cluster Skew, CS} = 1 + \text{abs}\left(\frac{\begin{matrix}\text{Count of websites} \\ \text{in cluster}\end{matrix} - \begin{matrix}\text{Avg count of websites} \\ \text{expected in each cluster}\end{matrix}}{\begin{matrix}\text{Avg count of websites} \\ \text{expected in each cluster}\end{matrix}}\right)$$

The values of cluster skew are greater than or equal to 1. A high value for cluster skew is undesirable, as it indicates that the number of websites in the cluster is either too less or too much. As a result, the overall score is penalized based on the cluster skew score. The website overlap score and the cluster skew score are used to score an individual cluster as follows:

$$\text{Cluster Score} = \frac{\text{Website Overlap}}{\text{Cluster Skew}}$$

Finally, a weighted average for all the individual cluster scores produced in an iteration of modelling is calculated that determines the quality of the cluster in the context of digital advertising. The final score derived for evaluating an entire iteration of the methodology is referred to as the cluster confidence index, CCI, and is calculated as follows:

$$\text{Cluster Confidence Index, CCI} = \frac{\sum_{i=1}^{k} \text{Cluster Score}_i \text{ x Total Websites}_i}{\sum_{i=1}^{k} \text{Total Websites}_i}$$

Where k corresponds to the total number of clusters generated in each iteration of clustering. The cluster confidence index ranges from 0 to 1. Intuitively, a higher CCI score corresponds to a cluster with websites very similar to each other, which is preferred. The CCI is the final evaluation criterion that will be used to evaluate the different hyperparameters and techniques implemented in the methodology to identify the best performing set of techniques for generating the website segments.

3.7 Summary

In this section, the methodology used in this study was explained in depth. The methodology discussed above was devised to help achieve the objectives that were stated in the first chapter of this study. The flow chart in figure 8 summarizes the entire methodology.

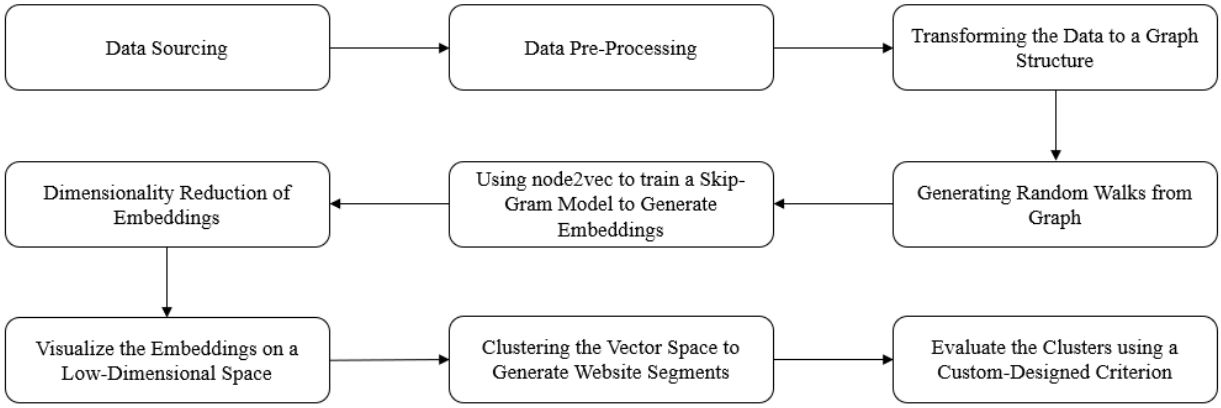


Figure 8. Flowchart of methodology

The chapter began by explaining how the data was sourced and the contents of the data. Post which, the issues pertaining to the dataset in its raw format were presented, along with the methods used for data pre-processing. Once the dataset was cleaned, the chapter discusses the data transformation process where the data in its tabular format is converted to a graph structure.

The chapter presents the node2vec technique used to generate website embeddings from the graph. Post generating the embeddings, dimensionality reduction techniques, and their purpose in this research have been discussed. The embeddings' dataset is then clustered, and the three different clustering techniques used in this study are discussed to produce a final set of clusters.

The chapter closes with the evaluation of the methodology used. A custom evaluation criterion has been designed to evaluate the techniques used in the methodology, known as the cluster confidence index (CCI). This metric will indicate which combination of techniques and hyperparameters is most suited for this dataset to generate segments of websites that would be used for contextual optimization of digital campaigns.

CHAPTER 4

ANALYSIS

The methodology discussed in the previous chapter explains the overall framework of the procedure implemented in this study. A significant aspect of the methodology discussed involved data preparation and exploratory analysis. The analysis presented in this chapter help give a better understanding of the kind of data that is being modelled upon and helps shape the implementation of the latter portions of the methodology.

4.1 Introduction

The chapter begins by discussing the steps taken in preparing the dataset. As discussed previously, the data was sourced in the JSON format and required an extensive level of pre-processing to get it ready for model building. The steps taken to clean the data while converting it into a tabular format are presented initially. The chapter then presents the exploratory data analysis done on the tabular data to extract meaningful insights that help shape the further sections of the methodology.

As part of the methodology, the tabular dataset is converted into a graph which is the eventual form of data modelled upon. The chapter presents an exhaustive overview of the structure of the graph being used, along with the necessary exploratory analysis done on the graph structure which helps develop an understanding of data that is being dealt with.

The final discussion in this section presents the architecture of the models that will be created. The previous chapter covered the different types of analytical algorithms and models that are implemented in this study. In this section, the actual setup of the models, along with the different hyperparameters to be tuned in each model, is discussed.

4.2 Data Preparation

Before dwelling into the exploratory analysis on the dataset, let's discuss the description of the dataset. The data sourced for this study has three columns of interest:

- The referrer website
- The destination website
- The count of users

All the unnecessary columns, such as the timestamp, were removed from the data, and it was aggregated appropriately to get a total count of users for each referrer-destination website movement.

As discussed in the methodology section, the data required a significant amount of pre-processing. The websites' columns required plenty of cleaning and sense-check to be done as they contained values with textual anomalies. Table 2 aims to summarize the steps taken in preparing the data before performing any type of exploratory data analysis on it.

Table 2. Data pre-processing steps

Column(s) Affected	Pre-processing Details
Timestamp	Column eliminated
Websites	Some values had an entire URL instead of just the domain name. The columns were made uniform to just hold the web domain.
Websites	Filtering out any punctuation that is not expected in the domain name of a website e.g. '%', '/', '\', '"', ';', '='
Websites	Filtering out websites that potentially hold explicit/illegitimate content
Websites	Filtering out websites that have observed less than 50 hits on the website
Websites	Filtering out anomalous entries where the referrer and destination websites are the same

4.3 Analyzing Trends in the Dataset

Before moving into applying the data-centric algorithms on the dataset, it is of utmost importance to develop an understanding of the data and the different trends that it exhibits. This section discusses the insights that could be driven from the tabular format of the data before applying the graph transformation on it.

Considering we're dealing with the data on websites, the most important aspect to understand is the distribution of hits on the different websites and how skewed it is. The box plot in figure 9 portrays what the distribution looks like for the hits per website.

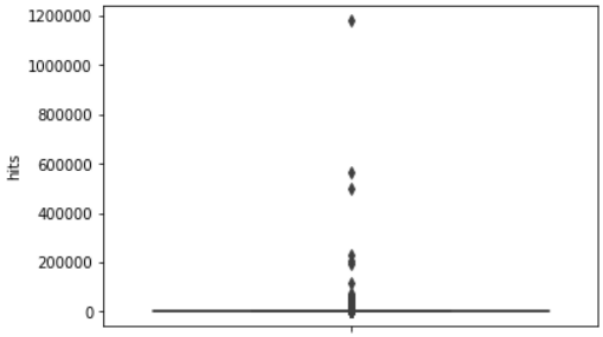


Figure 9. Distribution of website hits

From the figure, it can be inferred that over 90% of the websites have hits less than 1000, however, some websites are acting as extreme outliers. This means that there are a few websites that are occupying the greatest share of hits in the data.

The plot in figure 10 shows the share of hits occupied by the top 10 websites in the dataset. From the figure, it becomes evident that three websites: google.com, youtube.com, and facebook.com occupy over 25% of all the user hits on the dataset. This trend is expected, considering that the user base that these websites possess is massive. The outliers, in this case, are therefore representative of what a typical user online behaviour would look like, and hence there is no removal or processing done for treating the outliers.

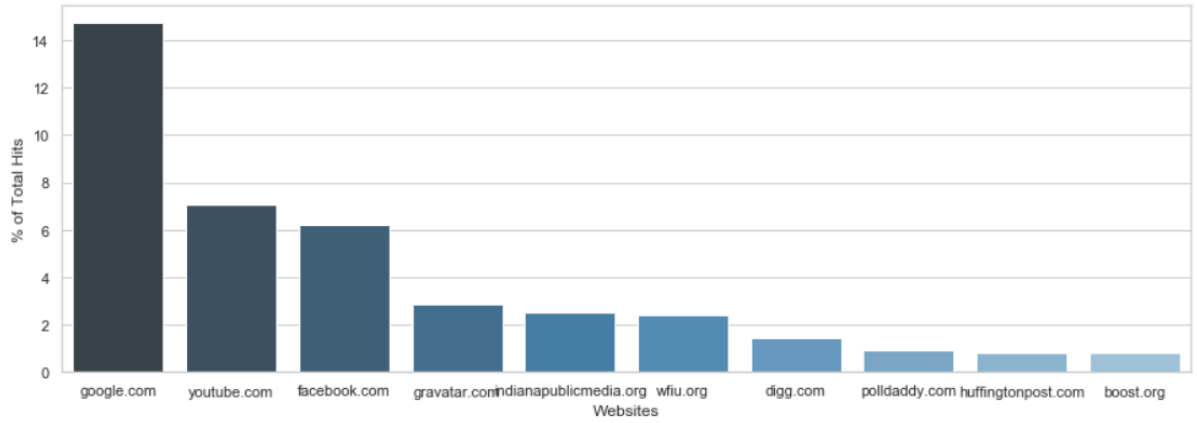


Figure 10. Websites with the greatest share of hits

Given the impact of outliers on the distribution of the data as shown in figure 9, the skewness of the data is not very evident. To get a better picture of how the data is distributed, a log-transform of the hits was used to visualize how the data points are skewed in the dataset. Figure 11 shows the distribution of the log-transformed data and gives a much clearer view of how the data is distributed.

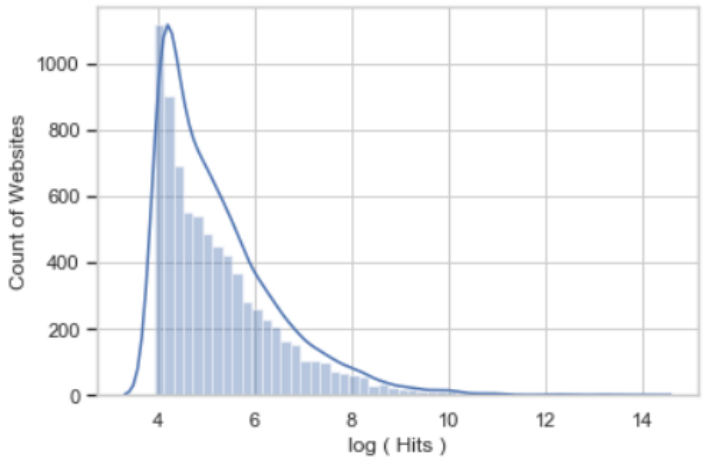


Figure 11. Distribution of website hits on a log scale

From figure 11, it can be perceived that the website hits variable follows a right-skewed distribution. This is a trend that would be expected in most samples of user browsing behaviour whereby there are plenty of websites with lesser hits, and only a few websites garnering the major share of hits.

The above analysis gives a clearer picture of what the distribution of the tabular data looks like. The data is then converted into a weighted graph structure, with the websites forming the nodes and the count of user shared by websites forming the weights of the edges.

4.4 Analyzing Trends in the Graph Structure

All the data-centric algorithms that have been finalized would be using the graph structure created from the tabular data as the input. This section of the analysis chapter discusses the insights that were derived from the graph structure created. Exploratory data analysis on graphs aims to capture different types of insights than tabular datasets, and the learnings from the analysis done here help in understanding the graph better.

4.4.1 Visualizing the Weighted Graph

To form the graph structure, the referrer and destination websites have been considered as the nodes of the graph, and the weights of the edges are representative of the count of users shared between the websites. To visualize what the graph looks like; figure 12 shows a small section of the entire graph.

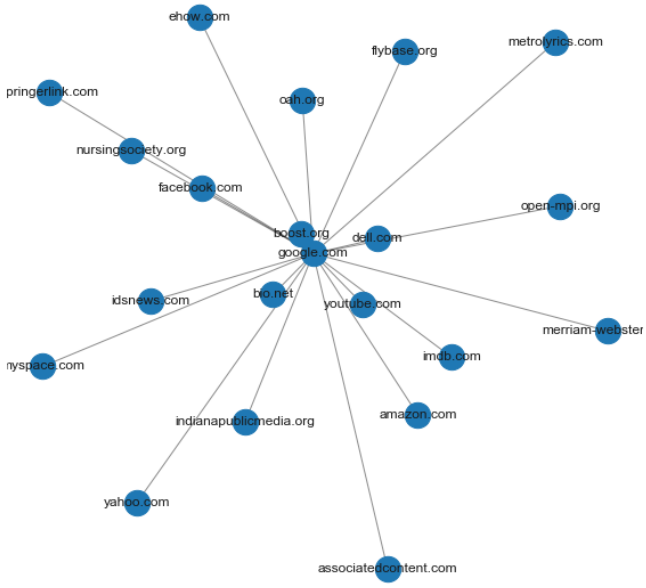


Figure 12. Weighted graph sub-section

The graph in figure 12 has been plotted considering google.com as the central node and all the nodes that are connected to it, with the lengths of the edges representative of the weights. The closer the nodes are to each other, the greater the users that they share.

4.4.2 Connectivity of the Weighted Graph

Graph structures can be characterized by the number of connections that are shared between the nodes (Li & Mao, 2012). A fully connected graph refers to a graph where all the nodes are connected to every other node in the structure. When working with real data, the chances of obtaining a fully connected data are minimal.

In general, graph structures from real data points may have nodes that are detached to different portions of the graph too. In such cases, one graph structure may have several connected subgraphs, however, these sub-graphs may not be connected.

Table 3. Details of connected sub-graphs

Number of nodes	Count of Connected Sub-graphs
2	543
3	66
4	28
5	13
6 - 17	11
5992	1

Table 3 shows the total number of connected subgraphs in the graph structure. It is observed that there are several isolated subgraphs in the structure with a very small number of nodes. There is one massive subgraph with a significant portion of the nodes in the dataset that are connected.

As part of this study, one of the objectives is to model the embeddings of the websites from the data. Embeddings essentially capture the relationships that exist between the different nodes in a graph in the form of a vector. As a result, the model must be trained on a connected graph, so that all the nodes exhibit some sort of a relationship with the other nodes taking into account their connection edges and edge weights. Therefore, all the smaller subgraphs have been eliminated, and the single connected sub-graph with the greatest number of nodes is considered.

4.4.3 Characteristics of the Weighted Graph

Graphs, such as the ones used in this study, are massive structures and hence it isn't feasible to visualize every section of the graph. However, considering that this graph structure will form the premise of every model that will be trained going forward, it is paramount to develop an understanding of the overall structure of it.

To develop this understanding, there is a certain set of characteristics exhibited by a graph that provides a picture of what the graph looks like. Table 4 provides the characteristics of the graph used in this study.

Table 4. Characteristics of the graph

Feature	Value	Feature Description
Number of Nodes	5,992	Total number of nodes in the graph
Number of Edges	8,412	Total number of edges in the graph
Graph Density	0.047%	The ratio of edges in the graph to the total possible edges
Graph Diameter	38,747	The shortest distance between the two furthest apart nodes

The graph density value in the table corresponds to the share of edges present in the data compared to all the possible edges if the graph was fully connected. The value indicates how well the nodes are connected and based on the value it is evident the nodes in the graph are sparsely connected.

The graph diameter in a graph corresponds to the shortest distance that exists between two of the furthest nodes. Every node in a graph has an eccentricity value, which corresponds to the distance between that node and the node furthest away from it. The graph diameter corresponds to the minimum value of all the eccentricities in a graph. In the context of the weighted graph, the graph diameter value of 38,747 corresponds to the maximum number of user website hits that connect any two websites in the data.

Along with the above features, there is another important feature known as node degree. The node degree of any node refers to the total number of edges connected to the node and is a very good indicator of how many connections every node in the graph has. Figure 13 shows the distribution of the node degrees for all the nodes in the graph.

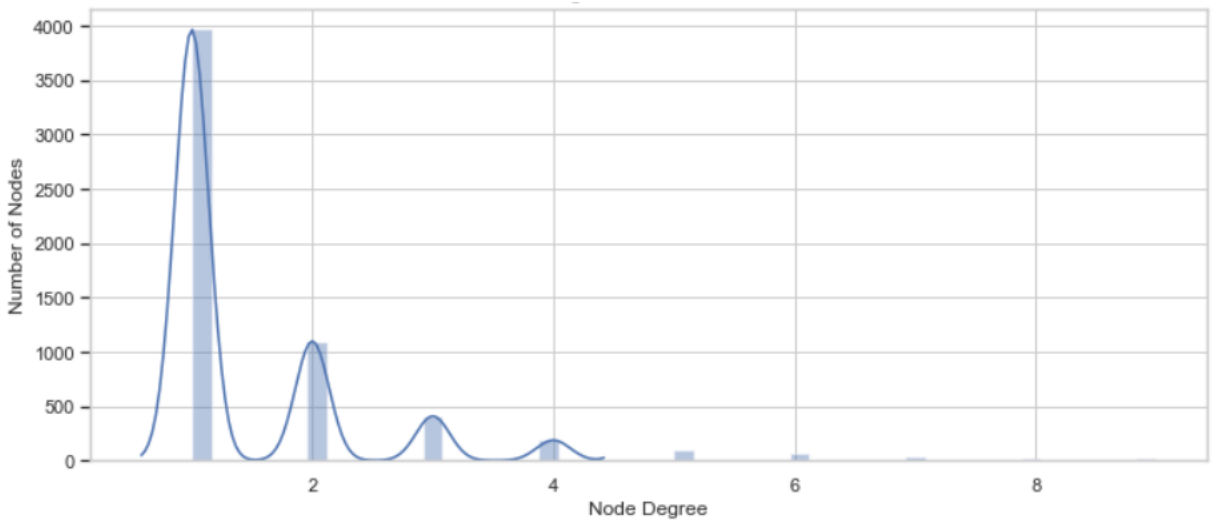


Figure 13. Node degrees distribution

From figure 13, it is evident that the majority of the websites in the graph have at most one connection with any other website in the data. This is also represented in the graph density score that was stated earlier and suggests that the nodes in the graph are connected to other nodes very sparsely.

4.4.4 Centrality of the Weighted Graph

Another key aspect of understanding complex graph structures is the centrality that is exhibited by the graph. The centrality of a graph refers to the presence of one or more nodes that carry the potential of influencing the graph massively as they have a really large number of connections, spanning to almost all the other nodes (Sharma, 2014). A graph with high centrality indicates that there is a single or a select few nodes that can be termed as the most important vertices and govern the relationships that exist between several other nodes.

Centrality in graphs can be identified using different algorithms. In this study, we've compared three separate centrality algorithms to identify the presence of any central nodes:

- Degree Centrality: Represents the number of nodes connected to any node. A node will exhibit high centrality if it has a large degree.
- Closeness Centrality: Represents the average length between a node and its shortest distance to all other nodes. A node will exhibit high centrality if it is very close to several other nodes.
- Betweenness Centrality: Represents the number of times any node has been traversed and forms part of the shortest distance between any two other nodes. A node will exhibit high centrality if it is likely to be a part of traversed between any two nodes.

The above-discussed centrality scores every node individually on how central they are in context to the entire graph. However, to identify the overall graph centrality score, the degree centrality can be extended to an entire graph as follows:

$$C_D(G) = \frac{\sum_n C_D^* - C_D(n)}{n^2-3n-2}$$

Where $C_D(G)$ represents the overall graph degree centrality, C_D^* represents the maximum degree centrality out of all nodes, and n represents the total number of nodes in the dataset.

The overall degree centrality of the graph in this study is 0.0062%. Such a low score indicates that there is minimal centrality being expressed by any single node in the graph. This is desirable as the model that will be built eventually aims to capture significant information from every node, and not let any single node dictate the relationships.

However, despite the minimal overall graph centrality, there does exist some nodes in the graph that exhibit more centrality than other graphs. Table 5 shows the top 10 websites exhibiting the highest centrality in the graph using the three different algorithms.

Table 5. Top node centralities

Degree Centrality	Closeness Centrality	Betweenness Centrality
google.com	google.com	google.com
youtube.com	facebook.com	youtube.com
facebook.com	youtube.com	facebook.com
gravatar.com	stumbleupon.com	gravatar.com
digg.com	digg.com	freep.com
stumbleupon.com	indianapublicmedia.org	rightcelebrity.com
projectwonderful.com	crunchgear.com	stumbleupon.com
assoc-amazon.com	techcrunch.com	digg.com
burstbeacon.com	sidereel.com	webmasterworld.com
bing.com	bing.com	speedycpm.net

From table 5, it can be inferred that the most central nodes out of all nodes are the websites: google.com, youtube.com, and facebook.com. This can be tied back to the analysis done in chapter 4.3 where it was identified that these three websites also have the largest share of hits for any website in the dataset.

However, since the overall graph centrality is minimal, the individual node centralities would not bias the vector representations of the nodes significantly, which is desirable.

4.5 Modelling Setup and Architecture

Having developed a broader understanding of the structure and the characteristics exhibited by the graph, decisions can be made regarding the selection of models and hyperparameters going forward. This section discusses how the learnings from the exploratory analysis done on the graph have been extended to preparing the setup of the model building phase of the methodology.

4.5.1 Node2Vec Setup and Architecture

The next step post running the analysis involves the generation of embeddings to represent the nodes of the graph on a vector space. As presented in the methodology, the process of generating embeddings broadly involves two steps:

- Generating random walks from the graph.
- Training a skip-gram model using the walks generated to produce a vector representation for each node.

The technique used for performing the tasks is node2vec, having evaluated the benefits that this technique provides over the other algorithms available. The node2vec algorithm can handle both the tasks together as discussed in the methodology chapter. However, the architecture to be implemented for the algorithm is one that is decided based on the objectives at hand, and the insights obtained from the exploratory analysis done above.

In chapters 3.3.2 and 3.3.3, the algorithm that node2vec follows was discussed, along with the implementation. The chapters discussed how the random walks are generated in node2ve (using BFS and DFS techniques), and the architecture of the skip-gram model used to generate the embeddings. Several hyperparameters were presented in those sections that govern the final architecture of the model created.

Table 6 depicts the set of hyperparameters that have been selected for the noe2ve model in this study, along with the justification of the choice.

Table 6. Node2vec setup and architecture

Hyperparameter	Value	Reason for Choice
Walk length	10	A walk of 10 nodes from each node captures the essence of every neighbourhood, whilst accounting for the limited computing power at hand for training the neural network
Number of Walks	100	A total of 100 walks are generated for every node, effectively capturing all possible nodes surrounding it
Return parameter, p	2	A value of p > 1 ensures that the walk doesn't traverse over the same node, hence increasing the diversity of nodes in a walk
In-Out parameter, q	2	A value of q>1 encourages exploration of the nodes closest to any node to capture local relationships (BFS)
Dimensions	10, 20, 50	The number of neurons in the single hidden layer, equivalent to the dimensions of the final embedding

The node2vec architecture demonstrated using the hyperparameters mentioned in table 6 has been used to generate the embeddings for the nodes in the graph. As evident from the table, the majority of the hyperparameters have been fixed as a result of the objective and the insights generated from exploring the graph structure. However, the hyperparameter governing the dimension of the embedding to be produced would be tested for different values. This study will be evaluating three different node2vec models with a different number of neurons in the hidden layer of the skip-gram model.

Post generating the embeddings, the next procedure is the dimensionality reduction of the vectors generated.

4.5.2 Dimensionality Reduction Setup and Architecture

Two dimensionality reduction techniques have been set up to reduce the dimensions of the embeddings generated i.e. PCA and t-SNE, one being a linear technique and the other being a non-linear technique.

Chapter 3.4.1 presents the architecture and working of the principal component analysis technique. For all the three sets of embeddings generated from the three skip-gram models, the vectors will be broken down into their respective principal components using this technique. Post that, the principal components that explain 90% of the variance in the dataset will be picked and used further in the procedure. Chapter 3.4.2 presents the architecture and working of the t-SNE technique. The chapter explained the optimization of a parameter known as the KL divergence. As part of the setup of the architecture in this study, several hyperparameters have been initialized as depicted in Table 7.

Table 7. t-SNE setup and architecture

Hyperparameter	Value	Reason for Choice
Perplexity	30	Refers to the number of nearest neighbours to be considered for the learnings, the choice of 30 is optimal considering the size of the dataset and the limited computing power at hand
Number of Iterations	50	Maximum number of iterations, 50 selected as it was enough to yield results for the gradient descent
Learning Rate	200	The rate at which gradient descent processes, the value 200 is observed to be adequate for the gradient descent to converge, and not get stuck on a local minima
Dimensions (n)	2, 5, 10, 20, 50	The dimension of the reduced embedded vector space
Distance Metric	Pairwise Euclidean	The Euclidean distance is used for computation of distances between data points

As evident from the table, the majority of the hyperparameters have been fixed as a result of the objective and the insights generated from exploring the graph structure. However, the hyperparameter governing the dimension of the final embedded vector space to be produced would be tested for different values. The KL divergence score for each technique would be evaluated, and the result that has observed the greatest convergence (minimum value for KL divergence) will be selected as the optimal technique.

4.5.3 Clustering Setup and Architecture

The final step in the methodology is the generation of clusters using three different techniques: k-means, hierarchical, and DBSCAN clustering. The choice of hyperparameters in this step becomes crucial as they govern the number and quality of clusters that are created, which is used to evaluate the entire methodology.

Chapter 3.5.1 presents the architecture of the k-means clustering technique. The algorithm used in k-means is discussed, along with the initialization of the centroids using k-means ++ to achieve the clustering output. Chapter 3.5.2 presents the architecture of the hierarchical clustering technique. The algorithm for hierarchical clustering, along with the specifics on the techniques used within the algorithm, such as the ward's linkage method, are discussed. Chapter 3.4.3 presents the architecture of the DBSCAN clustering technique. The algorithm used in the DBSCAN technique is discussed, and the various hyperparameters that are involved in controlling the algorithm, are presented.

The choice of hyperparameters opted for each of the technique have been finalized taking into consideration the number of records in the data, the computation resources available at hand and the purpose that is to be served by the eventual output. Table 8 presents the architecture of each of the clustering techniques in the form of the hyperparameter values selected, along with an explanation of the reason for the choice.

Table 8. Clustering setup and architecture

Clustering Technique	Hyper-parameter	Value	Reason for Choice
K-means	Maximum Iterations	50	Refers to the number of times the algorithm will iterate to generate clusters; 50 iterations were deemed good enough for convergence
	Number of Clusters	[5 to 50] in steps of 5	Identify the most optimal value by tuning this hyperparameter
	Cluster Initialization	k-means++	As discussed in chapter 3.5.1
Hierarchical	Strategy	Agglomerative Clustering	As discussed in chapter 3.5.2
	Linkage Criterion	Ward's	As discussed in chapter 3.5.2
	Number of Clusters	[5 to 50] in steps of 5	Identify the most optimal value by tuning this hyperparameter
	Affinity	Euclidean	The Euclidean distance has been opted for computing the linkage
DBSCAN	Eps, ε	[0.1, 0.2, 0.5, 1, 2, 5]	Identify the optimal radial distance around every point for clustering by tuning this hyperparameter
	Minimum Samples	[3, 5, 10, 20, 50, 100]	Identify optimal value for the minimum number of data points in a cluster by tuning this hyperparameter

The choice of architecture for each of the different modelling algorithms explained above was implemented and compared to evaluate the final set of clusters. This setup ties back to the objectives that were set up in chapter 1 and aim to achieve all of them.

4.6 Summary

During the study, an overall understanding of the data played a vital role in deciding the type of architecture to be used, the hyperparameters to be tuned, and other nuances for building out the models. This chapter covered the exploratory analysis done on the dataset and how it fed into the eventual design of the model architecture.

Chapter 4.2 introduced the dataset that has been used in the study and presented the pre-processing steps that were mandatory to get the data ready for modelling. This was followed by extracting the trends that were depicted from the tabular data, which helped provide a broad overview of the data that is being worked upon.

Chapter 4.4 discusses the most vital pieces of exploratory analysis, that eventually shaped up the architecture to be used for every model that follows. The section discusses the conversion of the tabular data into a graph structure. Understanding a graph can be very tedious, considering its massive size and numerous connections between the nodes. Exploratory analysis done on the graph involved extracting information such as graph statistics, graph connectivity, node degree distribution, node centralities, among other vital characteristics exhibited by the graph structures that help in getting an intuitive understanding of what the graph looks like

The learnings from the exploratory analysis on the graphs were fed into the architecture of the models that were implemented in the methodology. This chapter closes with a discussion on the different architectures that have been used, in terms of model hyperparameters, for achieving the objectives that were set out.

CHAPTER 5

RESULTS AND DISCUSSIONS

Post implementing the methodology discussed in chapter 3 and designing the models based on the architecture discussed in chapter 4, the results of the study are presented in this chapter. The results for the different models are evaluated to identify the best performing set of models.

5.1 Introduction

This chapter begins by presenting the results for all the individual sets of models implemented in the study. Chapter 4.5 discussed the sets of models that would be implemented, and the results from each are presented and evaluated against each other in the first section of this chapter.

The CCI score discussed in chapter 3 is used to evaluate these models and the best set of models is identified. The chapter further dwells into the specifics of this model and how it's results can be visualized and tied back to the optimization of digital ad campaigns. Specifically, a visual demonstration of how well the site domains have been clustered, along with checking the relevancy of the vector representations for each website.

The chapter closes with a discussion on how this study and the results obtained from it aims to fill the gaps in the digital advertising domains that were identified post the literature review presented in chapter 2.

5.2 Model Evaluation

Based on the architecture explained in the previous chapter, the project was set up to test the different models by varying the hyperparameters in the base models used. The scoring technique i.e. cluster confidence index, which was presented in chapter 3.6, has been used to evaluate the different sets of models to identify the set that best fits on the dataset being worked upon.

Table 9 presents the exhaustive list of sets of models implemented, along with the score for each of the sets. Using this table, the set of models having the greatest CCI score is deemed to be the most optimal set and outperforms all other models in the context of this study.

Table 9. Model scoring and evaluation

Node2Vec Dimensions	Dimensionality Reduction Technique	Clustering Technique	Optimal Number of Clusters	CCI Score
10	None	K-Means	50	0.286
		Hierarchical	50	0.305
		DBSCAN	2	0.410
	PCA	K-Means	40	0.327
		Hierarchical	50	0.285
		DBSCAN	5	0.200
	t-SNE	K-Means	45	0.381
		Hierarchical	50	0.388
		DBSCAN	1	0.000
20	None	K-Means	50	0.295
		Hierarchical	40	0.372
		DBSCAN	2	0.375
	PCA	K-Means	40	0.335
		Hierarchical	50	0.336
		DBSCAN	2	0.403
	t-SNE	K-Means	45	0.433
		Hierarchical	50	0.383
		DBSCAN	1	0.000
50	None	K-Means	50	0.260
		Hierarchical	50	0.340
		DBSCAN	5	0.000
	PCA	K-Means	35	0.327
		Hierarchical	50	0.340
		DBSCAN	2	0.379
	t-SNE	K-Means	45	0.368
		Hierarchical	50	0.351
		DBSCAN	1	0.000

From table 9, it can be inferred that the set of models that scored the highest is the 20 dimension node2vec model, followed by the t-SNE dimensionality reduction technique, and finally, the k-means clustering technique to generate the eventual segments of websites that can be used for the optimization of digital ad campaigns.

The results obtained give significant insights into the kind of data that is being modelled, the quality of the embeddings generated, and the kind of variance being explained by the different dimensions of the embedding vector. It's observed that the nodes in the graph structure used are optimally explained by a 20-dimensional vector representation.

Post this, it is observed that there is a need for dimensionality reduction, which explains that the embeddings generated exhibit some sort of multicollinearity. The most optimal dimensionality reduction technique is identified to be the non-linear one i.e. t-SNE. This is evident in figure 14 that compares the average result of all the clusters generated using these 2 techniques. The results in figure 14 only compare the outcomes from the k-means and hierarchical clustering techniques.

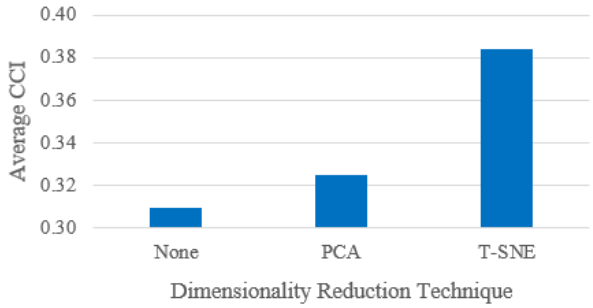


Figure 14. Comparison between dimensionality reduction techniques

Intuitively, it is known that PCA aims to preserve the global structure of the data by computing principal components, whereas, t-SNE aims to preserve the local structure of the data by ensuring that the data points close to each other in higher dimension are close to each other in the lower dimension too. This intuition is verified by the scoring technique used as t-SNE scored more than PCA in terms of dimensionality reduction, considering that the scoring technique defined in this study gives weightage to local connections in the graph.

The best performing method of clustering is k-means clustering. This Euclidean distance-based clustering technique identified 45 sets of clusters as the most optimal number of clusters. As a result, the study has produced 45 segments that consist of websites that share a common user base and can be used for targeting purposes in digital ad campaigns.

One important observation from the different sets of models is that the DBSCAN clustering technique was the least performing one compared to every other linear clustering technique. This is a clear indication of the linearity exhibited in the dataset post generating the embeddings and performing t-SNE to reduce the dimensions, hence the density-based clustering algorithm fails to segment the websites successfully.

5.3 Website Embeddings Evaluation

From the results evaluated in the above section, it is evident that the 20-dimensional embeddings from the node2vec model generated the most optimal vector representation of the websites from the graph structure. The embeddings are expected to be generated in a manner such that the vectors for websites relevant to each other are close to each other on the vector space.

To understand the quality of the embeddings generated, two tests were carried out:

- For a specific website, the n closest websites on the vector space were calculated and compared to check if they are similar to each other.
- A selected group of websites from several distinct website categories was visualized to identify whether they are separable. To visualize the websites, a 2-component t-SNE was implemented which allows the websites to be visualized on a dual-axis.

To perform the first task, the website selected was “economist.com”. The “economist.com” website is an e-magazine that discusses economic, financial, and political news across the globe. To evaluate the quality of the embeddings, 11 websites with an embedding vector closest to the vector representing “economist.com” were computed.

Figure 15 represents these websites in the form of a word cloud.

Figure 15. Websites with embeddings closest to “economist.com”

The size of the website in the cloud is representative of the cosine similarity that exists between the website and “economist.com”; marketwatch.com had the greatest cosine similarity of 0.96. From the results in Figure 15, the assessment for the quality of embeddings generated is quite positive. The website embeddings closest to the economist website are all related to economic and finance categories of websites, hence fall in a bracket similar to the economist website. This is a fair indication that the node2vec model implemented on the dataset has done a satisfactory job in representing the relationships between the websites on a vector space.

To help get a deeper understanding of the quality of the output produced by the models, the 2nd task stated above was carried out. Here, a set of websites belonging to five separate website categories, i.e. news blogs, business/finance/economic blogs, sports blogs, music blogs, and tech blogs, were picked from the dataset and their embeddings were fetched post implementing the 20-Dimension node2vec model.

A t-SNE dimensionality reduction algorithm was implemented to bring the embeddings down to two dimensions allowing them to be visualized on a 2-D axis. The components were then scaled between 0 to 1 to allow for comparison to be made between the websites.

Figure 16 shows the representation of these websites on a vector space. The website points on the scatter plot in the figure have been colour-coded to represent the website category that each website belongs to, as represented in the legend.

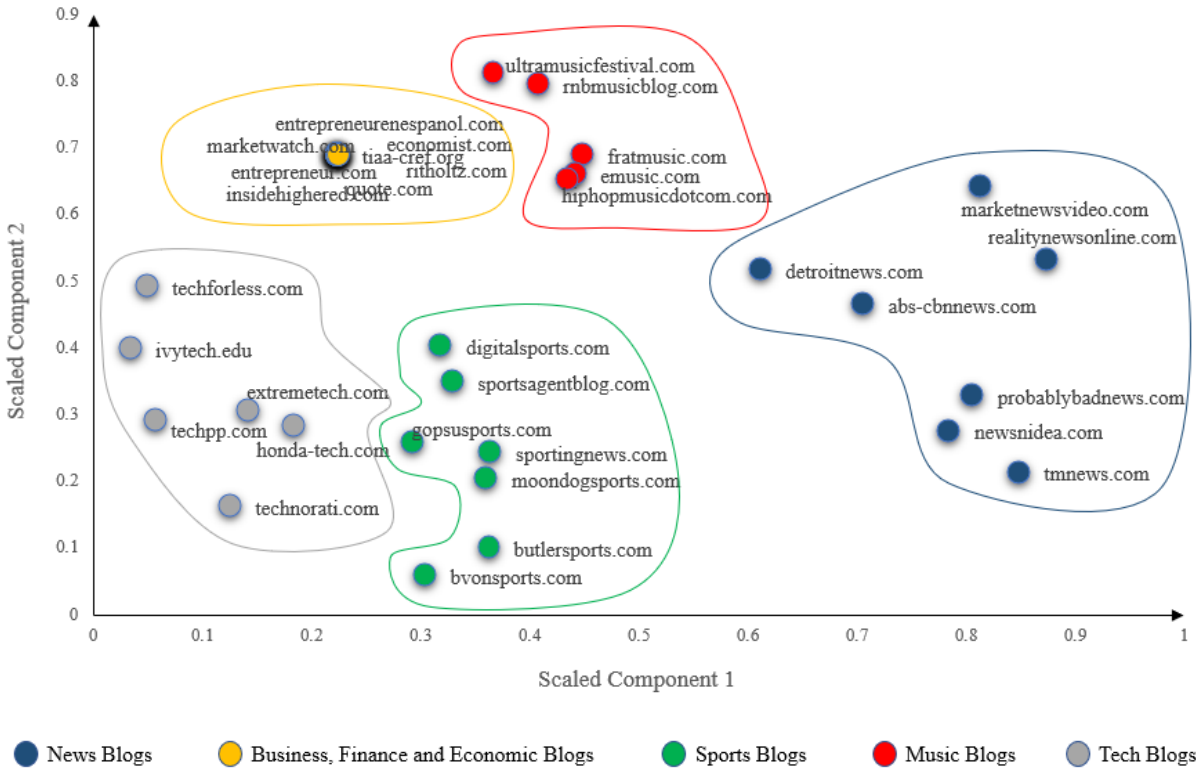


Figure 16. Visualizing the quality of embeddings

From the figure, it is evident that the algorithm does a really good job of mapping the websites on vector spaces with similar websites mapped closer together. The different clusters of websites belonging to similar categories have been shown as clusters on the graph and each website category seems to have found its territory in the vector space.

The two tasks presented in this section give a visual intuition of the embeddings that have been generated from the model and how well they can represent the graph structure on a vector space. The next section presents these results in the context of digital advertising, and how crucial a role they play in digital ad targeting and optimization.

5.4 Results in the Context of Digital Advertising

The title of this study mentions the use of embeddings to optimize digital ad campaigns. This section discusses how the results generated from the above set of models are relevant and can be activated upon, in the context of digital advertising.

Post performing the methodology, the websites were represented on a vector space in a manner such that the relationships existing between them were captured on the space. The final process of clustering resulted in 45 sets of websites. Each of these clusters consists of websites that share a very common user base. Therefore, each of these groups of websites can be targeted as separate segments for different advertisers based on the relevancy towards specific clusters.

For example, consider an advertiser that has launched a new pair of sneakers and is willing to reach out to potential buyers by advertising over the internet. Using the results produced in this study, specific clusters with websites relevant to the advertiser, such as websites with sports or clothing enthusiasts, can be selected and ads can be served on those websites. As a result, the advertiser can optimize their ad spend on websites that are relevant to their product.

Within the digital advertising space, the data points of users moving from one website to another can be collected covering a massive user base. As a result, the amount of data that can be fed into this methodology would be a lot more than the data sourced for this study. The methodology proposed above can be used to capture the relationships between websites across all the users on the internet and cluster them effectively. This allows an advertiser to reach their relevant audiences, even in the cookie-less world.

5.5 Discussion

The literature review presented in chapter 2 discussed the landscape of digital advertising and several limitations were presented pertaining to the government regulations on cookie-based targeting and the lack of contextual targeting strategies at any advertiser's disposal. This study proposes a new strategy that can contribute to filling this gap.

Digital advertising would be deemed effective only when the ads have been served to the audience relevant to the advertiser. Given how large the audience size is on the internet, ad spends would inevitably be wasted by showing ads to users who aren't relevant to the advertiser brand if the targeting is not optimized. As presented in the literature review, the existing techniques for defining targeting strategies are majorly based out of prior campaign performance, or the intuition of the advertiser.

This study follows a novel approach for filling the gap, by introducing the concept of graph-based embeddings to solve the problem. This technique is heavily driven by data, as it attempts to capture the online behaviour of users on the internet to classify websites into clusters. The modelling techniques implemented in this study would enable advertisers to channel their ad spends effectively on the clusters of websites most relevant to their desired audience.

5.6 Summary

This chapter presented the results derived from the study in order to achieve all the objectives defined in chapter 1. It began by scoring the different sets of hyperparameters that were trained on the graph structure to narrow down on the most optimal set of models.

Having identified the best set of hyperparameters to achieve the objectives, they were implemented on the dataset to generate the embeddings and eventually generate the cluster of websites. To represent the effectiveness of the embeddings, Chapter 5.3 presented the n-closest websites to a selected website based on Euclidean distance to intuitively show the relevancy of the websites that are closer together on the vector space.

Chapter 5.4 presents the embeddings generated for a selected set of websites on a 2-dimensional vector space to give a visual understanding of how well the models have managed to separate the different clusters of websites based on the domain category. The chapter then moves further to tie back the results to the digital marketing domain. The discussion dwells upon how the methodology in this study can be leveraged to fill the existing gaps in contextual ad targeting strategies.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

This chapter provides the closing remarks on the study done. It broadly covers the conclusion from the study, the limitations that needed to be curbed, and the future scope of the work presented in this thesis.

6.1 Introduction

Contextual ad targeting and optimization techniques have begun to see a lot of traction, and the methodology presented in the study so far goes a long way in contributing to this field. Having discussed the entire methodology, the analysis, and the results produced in this study; this final chapter of the thesis presents the closing remarks.

The chapter begins with re-iterating the objectives that were set out to be achieved initially and how the methodology and results were able to justify the completion of each objective. This is followed by discussing the limitations and constraints that were faced throughout the implementation of the methodology.

Further, the chapter presents the contribution and impact that the results from this study are desired to have on the existing knowledge in the domain and the research community. The chapter ends with presenting the potential that the study carries and how it can be further built upon in the future.

6.2 Discussion and Conclusion

In this section, the objectives that were derived in chapter 1 are reviewed individually. Each objective is critically analyzed, and a discussion is presented on how they were achieved over the course of the entire implementation of this study.

- To represent the websites in the form of a graph structure

The initial phase of the methodology involved the sourcing of the data and cleaning it to form a refined tabular dataset containing three desired columns; the referrer website, the destination website, and the count of users following the path. This dataset was converted into a graph structure using the network library, as presented in chapter 3.2.3. As a result, the eventual representation of the dataset was a weighted graph structure where the nodes represented the websites and the edges represented the number of users shared by any two websites.

- To model website embeddings from the graph using geometric deep learning techniques

The literature review presented in this research discussed the various methods of generating embeddings from a graph structure. In this study, the node2vec technique was implemented to generate embeddings. Chapter 3.3 presents the implementation of this technique whereby random walks were generated from the graph structure that served as an input to a skip-gram model whose weights were used to derive the embeddings. As a result, the websites could be represented on a vector space, such that the relationships between any two websites would be reflected in form of the Euclidean distance between those two websites on the vector space. The results have also been visualized on a 2-D vector space in chapter 5.3.

- To generate clusters of relevant websites using the embeddings

This objective involved performing segmentation of the websites into relevant buckets that can be used for targeting in ad campaigns in the form of website whitelists. Chapter 3.5 presents the clustering techniques that were implemented in the study to achieve this objective. Having generated the vector representation of the websites, the data was tested for dimensionality reduction that would handle any multicollinearity exhibited between dimensions of the vector representation. Post that, three different clustering algorithms were implemented on the data to identify the algorithm that produces the optimal set of website clusters.

- To design a relevant criterion and evaluate the various techniques used in the methodology to identify the best performing technique

This objective involved designing a new criterion that would be used to evaluate the methodology. The criterion had to be designed in a manner such that the score is reflective of the application that the final output is being created for. Chapter 3.6 presents the CCI score that was designed for evaluation purposes. The evaluation results for the various iterations of the methodology using different sets of models are presented in chapter 5.2. Using this result, the best performing set of models were identified and implemented on the dataset to produce the final set of clusters.

6.3 Limitations

Throughout the methodology, several limitations and constraints had to be curbed to achieve the final set of results. The limitations, in some ways, also influenced the architecture of the models used in the methodology. This section presents these limitations, and the impact they had on the study.

Any deep learning algorithm requires immense computational resources to execute when training a model. This computational power requirement becomes even more essential as the size of the data being trained increases. Given the limitation of computational resources available for this study, in terms of memory available and processor power, the architecture presented in chapter 4.5 was designed in a manner that can produce the desired output over the finite duration of this study.

The dataset used in this study was sourced from an open-source platform. It was assumed that the dataset would be reflective of any user's online behaviour and has been used to implement the methodology. Given that the data was only reflective of traffic observed in a university over a month, the results may be different when considering traffic data from different regions. However, the methodology proposed in this study can be extended to web traffic data from any source.

6.4 Contribution to Knowledge

The literature review discussed the existing scenario of contextual targeting in the digital advertising landscape and presented the gaps that exist in this domain considering the lack of data-driven techniques for coming up with targeting strategies. The methodology proposed in this study attempts to fill this gap by coming up with a predictive data-backed approach of generating segments of websites that can be used for ad targeting an optimization in the form of website whitelists.

The geometric deep learning techniques discussed in this study have seen applications in some very niche domains. Word2vec technique for producing word embeddings is a vital cog in the natural language processing space and has found applications in fields such as text classification, sentiment analysis, etc. The graph-based adaption of this technique i.e. node2vec, that has been implemented in this study, has seen applications in graph structures primarily in link prediction problems or graph visualization problems. The application for node2vec proposed in this study is one that has not been explored before in the research community.

The application of node2vec in digital ad campaign optimization done in this study can serve as the foundation for other researchers to expand upon. The next chapter presents the potential that this research carries and how it can be embellished further to find other relevant use-cases. Researchers can leverage this study to get a deeper understanding of the modelling architecture used and delve deeper into the field to find other useful solutions in the field of digital advertising.

The scoring criterion presented in this thesis is designed to cater to the digital ad campaign use-case, and the score for the methodology is reflective of the expected performance of the clusters in the digital ad campaigns. This scoring criterion is novel and can be used as a benchmark for research in any sort of clustering study on the contextual elements of digital advertising data.

6.5 Future Recommendations

This study is confined to generating the embeddings for websites and applying them in the space of digital advertising for optimizing campaigns. However, as mentioned in chapter 1.4, several other applications are directly achievable from the output of the methodology in this study.

The embeddings generated in this study are based on data scoped from the web traffic data made available by Indiana University. When implementing this for any specific advertiser, a much larger subset of data can be utilized under the same methodology to generate these embeddings. Since the results are entirely driven by the relationships that exist between the websites, a richer dataset would enhance the quality of the output as it would be able to capture many more associations between websites.

The application discussed in this study involves using the embeddings to generate clusters of websites that can be targeted in ad campaigns using website whitelists. In addition to this, the embeddings can be used for other purposes too, for example:

- Predicting the next website that a user would visit based on their web history: Based on the type of websites that a user has visited, a vector score can be generated to identify where the user falls in the entire embedding space. Using this, a distance metric can be used to identify the websites that the user is likely to visit next.

- Using the website embeddings in machine learning problems that have websites as one of the predictor variables: Under usual circumstances, the website variable in any prediction problem would be one-hot encoded or target encoded to convert them from categorical to a continuous variable. Instead of these naïve approaches, the website variable can be replaced with their vector representation generated from the node2vec algorithm, which would be a heavily data-driven technique of encoding and also help capture the relationships that exist between websites in the variable.

The technique used for generating the embeddings in this study is node2vec. Other geometric deep learning techniques can be leveraged to perform this task, as presented in the literature review in chapter 2.9. In the future, the clusters generated can be compared using the different deep learning techniques to identify the technique that performs the most optimal task.

The embeddings generated in this study are entirely based on interactions between websites in terms of the number of users that are shared between them. As a result, only website-website relationships are being captured here. This can be enhanced by capturing user-website and user-user relationships too. As a result, the graph structure would be a bipartite graph where the users and websites would be two independent sets of nodes. Training a node2vec model on this would result in representing both the users and the websites on a vector space, which would enable advertisers to not only identify websites that are similar to each other, but also users that are similar to each other in terms of their browsing behaviour, and the type of websites each user generally visits.

6.6 Summary

This chapter provided the closing remarks to the thesis and the study done. The chapter began with re-iterating the objectives that had been set in chapter 1 and how they were achieved throughout the study. Along with that, the limitations that existed while executing the methodology was presented.

The chapter then discusses the impact that this study would have in the digital advertising domain, along with the contribution towards the research community. Finally, the section ponders upon the future prospects of the study done and presents a comprehensive list of applications and use-cases that researchers can leverage this study to expand further on the topic.

REFERENCES

Papers

Abiodun, O.I., Jantan, A., Omolara, A.E., Dada, K.V., Mohamed, N.A.E. & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*. [Online]. 4 (11). Available from: https://doi.org/10.1016/j.heliyon.2018.e00938.

Adolfsson, A., Ackerman, M. & Brownstein, N.C. (2019). To cluster, or not to cluster: An analysis of clusterability methods. *Pattern Recognition*. 88. p.pp. 13–26.

Agarwal, P. & Shukla, V.K. (2013). E-marketing Excellence: Planning and Optimizing Digital Marketing. *International Journals of Marketing and Technology*. 3 (March 2017). p.pp. 131–133.

Arthur, D. & Vassilvitskii, S. (2007). k-means++: The Advantages of Careful Seeding. *Proceeding of the 18th Annual ACM-SIAM Symposium of Discrete Analysis*. p.pp. 1027–1035.

Bala, M. & Deepak Verma, M. (2018). A Critical Review of Digital Marketing. *International Journal of Management*. 8 (10). p.pp. 321–339.

Berry, S. (2019). *What Is Contextual Targeting? (And Why Does It Even Matter?)*. [Online]. 2019. Available from: https://www.webfx.com/blog/marketing/contextual-targeting/.

Blum, A., Wardman, B., Solorio, T. & Warner, G. (2010). Lexical feature based phishing URL detection using online learning. *Proceedings of the ACM Conference on Computer and Communications Security*. p.pp. 54–60.

Bussche, D.A.F. von dem & Voigt, P. (2017). *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer International Publishing.

Choi, H., Mela, C.F., Balseiro, S. & Leary, A. (2019). Online Display Advertising Markets: A Literature Review and Future Directions. *SSRN Electronic Journal*.

Deshpande, I. (2019). *What Is Programmatic Advertising? Definition, Types, Channel, and Advantages*. [Online]. 2019. Available from: https://www.martechadvisor.com/articles/ads/what-is-programmatic-advertising/.

Enberg, J. (2019). *Global Digital Ad Spending 2019*. [Online]. 2019. Available from: www.emarketer.com/content/global-digital-ad-spending-2019.

García, J.J.L., Lizcano, D., Ramos, C.M.Q. & Matos, N. (2019). Digital marketing actions that

achieve a better attraction and loyalty of users: An analytical study. *Future Internet*. 11 (6). p.pp. 1–16.

Goldfarb, A. & Tucker, C. (2011). Online Display Advertising: Targeting and Obtrusiveness. *Marketing Science*. 30 (3). p.pp. 413–415.

Gonzalvez-Cabañas, J.C. & Mochón, F. (2016). Operating an Advertising Programmatic Buying Platform: A Case Study. *International Journal of Interactive Multimedia and Artificial Intelligence*. 3 (6). p.p. 6.

Grover, A. & Leskovec, J. (2016). node2vec: Scalable feature learning for networks. *the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. p.pp. 855–864.

Hagberg, A.A., Schult, D.A. & Swart, P.J. (2008). Exploring network structure, dynamics, and function using NetworkX. *7th Python in Science Conference (SciPy 2008)*. (January 2008). p.pp. 11–15.

Hamilton, W.L., Ying, R. & Leskovec, J. (2017). Representation Learning on Graphs: Methods and Applications. *IEEE Data Eng. Bull.* [Online]. 40. p.pp. 52–74. Available from: http://arxiv.org/abs/1709.05584.

Hasan, M. & Zaki, M. (2011). A Survey of Link Prediction in Social Networks. *Social Network Data Analytics*. (March 2011). p.pp. 243–275.

Hofgesang, P.I. & Kowalczyk, W. (2006). Analysing clickstream data: From anomaly detection to visitor profiling. *Belgian/Netherlands Artificial Intelligence Conference*.

Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R. & Wu, A.Y. (2002). An Efficient k-Means Clustering Algorithm: Analysis and Implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 24. p.pp. 881–892.

Klein, S. (2010). *Introduction to Digital Advertising on Google, Facebook, & More*. [Online]. 2010. Available from: https://www.artonicweb.com/learn/digital-advertising/.

Le, H., Pham, Q., Sahoo, D. & Hoi, S.C.H. (2018). URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection. *ACM Conference, Washington, DC, USA*. [Online]. Available from: http://arxiv.org/abs/1802.03162.

Li, R.H., Yu, J.X., Qin, L., Mao, R. & Jin, T. (2015). On random walk based graph sampling. *Proceedings - International Conference on Data Engineering*. 2015-May (April). p.pp. 927–938.

Li, X. & Mao, Y. (2012). *A survey on the generalized connectivity of graphs*. [Online]. (February 2014). Available from: http://arxiv.org/abs/1207.1838.

Lin, M.S., Chiu, C.Y., Lee, Y.J. & Pao, H.K. (2013). Malicious URL filtering - A big data application. *Proceedings - 2013 IEEE International Conference on Big Data, Big Data 2013*. p.pp. 589–596.

Maaten, L. van der & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*. [Online]. 1. p.pp. 1–48. Available from: http://www.cs.toronto.edu/~hinton/absps/tsne.pdf.

Marouchos, C. (2020). *Programmatic Advertising 101: Campaign Optimization*. [Online]. 2020. Available from: https://www.centro.net/blog/programmatic-101-campaign-optimization.

Meiss, M., Gonçalves, B., Ramasco, J.J., Flammini, A. & Menczer, F. (2010). Modeling Traffic on the Web Graph R. Kumar & D. Sivakumar (eds.). *Kumar R., Sivakumar D. (eds) Algorithms and Models for the Web-Graph. WAW 2010. Lecture Notes in Computer Science*. [Online]. 6516 (May 2014). Available from: http://link.springer.com/10.1007/978-3-642-18009-5.

Meiss, M.R., Menczer, F., Fortunato, S., Flammini, A. & Vespignani, A. (2008). Ranking web sites with real user traffic. *WSDM'08 - Proceedings of the 2008 International Conference on Web Search and Data Mining*. (October 2014). p.pp. 65–75.

Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013a). Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*. p.pp. 1–12.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. & Dean, J. (2013b). Distributed Representations of Words and Phrases and their Compositionality. *Advances in Neural Information Processing Systems*. (October).

Mishra, S., Sarkar, U., Taraphder, S., Datta, S., Swain, D., Saikhom, R., Panda, S. & Laishram, M. (2017). Multivariate Statistical Data Analysis- Principal Component Analysis (PCA). *International Journal of Livestock Research*. [Online]. (January). p.p. 1. Available from: http://www.ejmanager.com/fulltextpdf.php?mno=261590.

Murtagh, F. & Contreras, P. (2011). Methods of Hierarchical Clustering. *ArXiv*. abs/1105.0.

Murtagh, F. & Legendre, P. (2011). Ward's Hierarchical Clustering Method: Clustering

Criterion and Agglomerative Algorithm. *ArXiv*. [Online]. (June). p.pp. 1–20. Available from: http://arxiv.org/abs/1111.6285%0Ahttp://dx.doi.org/10.1007/s00357-014-9161-z.

Naik, G.R. (2018). *Advances in Principal Component Analysis*. 1st Ed. Springer Singapore.

Perozzi, B., Al-Rfou, R. & Skiena, S. (2014). DeepWalk. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14*. [Online]. 2014, New York, New York, USA: ACM Press, pp. 701–710. Available from: http://dl.acm.org/citation.cfm?doid=2623330.2623732.

Ram, A., Jalal, S., Jalal, A.S. & Kumar, M. (2010). A Density Based Algorithm for Discovering Density Varied Clusters in Large Spatial Databases. *International Journal of Computer Applications*. 3 (6) p.pp. 1–4.

Sajjad, H.P., Docherty, A. & Tyshetskiy, Y. (2019). *Efficient Representation Learning Using Random Walks for Dynamic Graphs*. [Online]. Available from: http://arxiv.org/abs/1901.01346.

Sarveniazi, A. (2014). An Actual Survey of Dimensionality Reduction. *American Journal of Computational Mathematics*. 04 (02). p.pp. 55–72.

Savić, D. & Veinović, M. (2018). Challenges of General Data Protection Regulation (GDPR). In: *Sinteza 2018*. 2018.

Sharma, P. (2014). Centrality Measures in Social Networking: Study and Analysis Using NetDraw 2.138 in UCINET6. *International Journal of Engineering Research & Technology (IJERT)*. 3 (4). p.pp. 2583–2586.

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J. & Mei, Q. (2015). LINE: Large-scale Information Network Embedding. In: *Proceedings of the 24th International Conference on World Wide Web - WWW '15*. [Online]. 2015, New York, New York, USA: ACM Press, pp. 1067–1077. Available from: http://dl.acm.org/citation.cfm?doid=2736277.2741093.

Vathy-Fogarassy, Á. & Abonyi, J. (2013). *Graph-Based Clustering and Data Visualization Algorithms*. 1st Ed. Springer-Verlag London.

Wang, D., Cui, P. & Zhu, W. (2016). Structural Deep Network Embedding. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. [Online]. 13 August 2016, New York, NY, USA: ACM, pp. 1225–1234. Available from: https://dl.acm.org/doi/10.1145/2939672.2939753.

Wang, F., Franco-Penya, H.-H., Kelleher, J.D., Pugh, J. & Ross, R. (2017). An Analysis of the

Application of Simplified Silhouette to the Evaluation of k-means Clustering Validity. In: *13th International Conference on Machine Learning and Data Mining MLDM 2017*. 2017.

Yasmin, A., Tasneem, S. & Fatema, K. (2015). Effectiveness of Digital Marketing in the Challenging Age: An Empirical Study. *The International Journal of Management Science and Business Administration*. 1 (5). p.pp. 69–80.

Zhang, Z., Zhao, J. & Yan, X. (2018). A web page clustering method based on formal concept analysis. *Information (Switzerland)*. 9 (9).

Books

Bussche, D.A.F. von dem & Voigt, P. (2017). *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer International Publishing.

Gron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 1st Ed. O'Reilly Media, Inc.

Naik, G.R. (2018). *Advances in Principal Component Analysis*. 1st Ed. Springer Singapore.

Vathy-Fogarassy, Á. & Abonyi, J. (2013). *Graph-Based Clustering and Data Visualization Algorithms*. 1st Ed. Springer-Verlag London.

Newspaper Articles, Online Articles and Websites

Berry, S. (2019). *What Is Contextual Targeting? (And Why Does It Even Matter?)*. [Online]. 2019. Available from: https://www.webfx.com/blog/marketing/contextual-targeting/.

Deshpande, I. (2019). *What Is Programmatic Advertising? Definition, Types, Channel, and Advantages*. [Online]. 2019. Available from: https://www.martechadvisor.com/articles/ads/what-is-programmatic-advertising/.

Enberg, J. (2019). *Global Digital Ad Spending 2019*. [Online]. 2019. Available from: www.emarketer.com/content/global-digital-ad-spending-2019.

Klein, S. (2010). *Introduction to Digital Advertising on Google, Facebook, & More*. [Online]. 2010. Available from: https://www.artonicweb.com/learn/digital-advertising/.

Marouchos, C. (2020). *Programmatic Advertising 101: Campaign Optimization*. [Online]. 2020. Available from: https://www.centro.net/blog/programmatic-101-campaign-optimization.

APPENDIX A

RESEARCH PROPOSAL

Generating Website Embeddings to Optimize Digital Ad Campaigns

Vyom Pankajkumar Bhatt

MSc. Data Science

Abstract

Neural networks have seen applications in a vast variety of domains. In this work, we explore how it's application can be extended to one of the less experimented domains, digital advertising. Digital ad campaigns can be optimized in a wide number of ways, from contextual targeting to audience-based targeting strategies. Here, we delve into how we can use neural networks to optimize digital ad campaigns by identifying the relevant websites for different advertisers. We propose using a graph-based representation for connections between websites, then using researched neural network techniques to represent these websites in a low-dimensional vector space by generating nodal embeddings. We evaluate these embeddings on a classification task to identify any incremental performance that it generates compared to traditional techniques. Following that, we investigate the applications that these embeddings have in the digital advertising space and how they can be used to optimize digital ad campaigns.

Key Words: website embeddings, graph embeddings, geometric deep learning, digital advertising

Table of Contents

1. Introduction

The digital advertising space has seen massive growth in recent times as the worldwide digital ad spend is expected to exceed $350 billion by the end of 2020, and over 50% of all ad spend would comprise of digital ads (eMarketer, 2019). Websites on the internet hold a major part of the ad inventory that is available for advertisers to reach their customers.

The major challenge for any advertiser, however, is to identify what websites should they serve ads on to ensure they're reaching their potential customer (Agarwal & Shukla, 2013), hence avoiding wastage of ad spend on irrelevant websites. Marketers need to constantly keep optimizing their ad campaigns to ensure they're spending on ad slots on websites that are likely to perform better. Current techniques of optimizing digital campaigns revolve around using intuition or historical ad campaign learnings to narrow down on the websites to serve ads on (WebFX, 2019). These have proven to be effective in cases, but the learnings cannot be used across different verticals of advertisers; learnings from, say, a retail advertiser campaign may not be relevant for another advertiser in the travel domain.

In this project, we follow a data-driven approach to perform this optimization, which is advertiser vertical agnostic. We first represent website visitation patterns of a set of users in the form of an undirected graph, with websites as the nodes and edges being the number of users common to both connecting websites. We then propose using researched geometric deep learning techniques (Grover & Leskovec, 2016) to generate nodal embeddings for the graph structure. These embeddings can be used to visualize the websites on a lower-dimensional vector space. We proceed to investigate how the learnings from these embeddings can then be extended to optimize digital ad campaigns. We explore the following applications of our embeddings:

1. Predicting the next website that a user visits based on their history
2. Using a clustering algorithm (J. Han and M.Kamber, 2000) to segment similar websites
3. Using the embeddings as a feature in classification problems like CTR prediction

2. Background and related research

Digital advertising is a broad field with different techniques, one of them being display advertising, which involves advertisers spending on banner ads on websites (Bala & Deepak Verma, 2018). We understand that optimization in digital campaigns can take various approaches, such as techniques to ensure customer retention and loyalty (García et al., 2019), or identifying relevant audience demographics for different advertisers, among other methods. Our work takes a more contextual approach to optimize digital campaigns by understanding what websites are suitable for different types of advertisements.

We take inspiration from the research done around methods and convenience of representing data in form of graphs for scientific computations (Hagberg et al., 2008). Post transforming our dataset into a graphical structure, we extract features using random walks to train a model that would generate embeddings for the website (Sajjad et al., 2019). The modelling process is an application of the node2vec technique (Grover & Leskovec, 2016) which was introduced as a framework for vector representations of nodes in a network.

There has been some research done around website classification and categorization before our work. Deep Learning techniques have been used to generate embeddings of websites by extracting their lexical features for malicious URL detection (Le et al., 2018). Work done around website clustering has previously also been done using formal concept analysis (Zhang et al., 2018). Graph-based learning problems have previously heavily been researched in social network domains (Hasan & Zaki, 2011) where graph features are extracted to predict links.

However, none of the work done previously can meet our objective of using website embeddings to optimize digital ad campaigns.

3. Aim and Objectives

The main aim of this research is to propose a new technique for the optimization of digital campaigns. We leverage learnings from previous research in the field of graph-based feature learning to generate embeddings of websites that can be used to optimize digital campaigns.

The research objectives are formulated based on the aim defined above and they are as follows:

- To generate website embeddings from our dataset
- To evaluate the performance of our website embeddings in a classification problem
- To investigate the applications of our website embeddings in optimizing digital ad campaigns

To investigate the application of our website embeddings in digital ad campaigns, we will be performing the following experiments:

- To predict the next website that a user would visit based on their browsing history
- To cluster the websites based on embeddings and intuitively check the quality of the clusters

4. Significance of the Study

The study done in this project contributes to the field of digital advertising as we're proposing a novel way of contextually optimizing digital ad campaigns. We further investigate methods on how our work contributes to the field of advertising using an industry-relevant dataset. The work from this project would be valuable to anyone working or performing research in the digital advertising or ad-tech domain.

This project also contributes to the applications researched in the field of geometric deep learning techniques performed on graph structures.

5. Scope of the Study

The work done is limited to the data available from web traffic requests for November 2009 (Meiss et al., 2010). Hence, the embeddings generated for the websites would only be reflective of the browsing behaviour of users during the timeframe capture in the dataset.

Training of the neural networks to generate nodal embeddings is a computationally heavy task, hence we have sampled the available dataset to ensure we have results to show in the stipulated timeframe.

The embeddings generated are limited to website-website interactions. This can further be extended to user-website interactions, which would intuitively give a much fairer representation of how similar certain websites are to each other. However, we've not considered user interactions due to the constraint in the time and resources available as part of the project.

6. Research Methodology

The data that we will use is a collection of web traffic requests for November 2009 (Meiss et al., 2010). The data is aggregated at an hourly level and contains 4 columns:

5. Timestamp: hour at which the data is aggregated at
6. From: this is the referrer website; the column contains the website from where a user originates
7. To: this is the target website; the column contains the website where the user goes to
8. Count: the number of users in the hour that follow the path from the referrer to target website

The data cited was originally in the JSON format, and it needs to be parsed and cleaned to be used as a data frame. Considering the scale of the data, as part of our work, we will be sampling the data to make it feasible for our aim, keeping in mind the resources that we have available at our hand.

Post cleaning the data, we will aggregate the entire data so that we have an overall count of users overlapping between two websites, and not at an hourly level. This will then be converted into a graph structure, such that the nodes of the graph correspond to the websites and the edges represent the count of users that they share.

Once we have the graph representation, we will use random walks to identify random paths from the graph by specifying certain node2vec hyperparameters. Having generated the random walks, we will train a skip-gram model on these walks. The skip-gram model is similar to the one used in the word2vec model (Mikolov et al., 2013a) that uses a neural network with one hidden layer and a softmax output layer. The essence of node2vec is to learn the representation of the nodes in a low-dimensional vector space in a manner such that the nodal neighbourhoods are being preserved. These vector representations of the websites are what we refer to as embeddings.

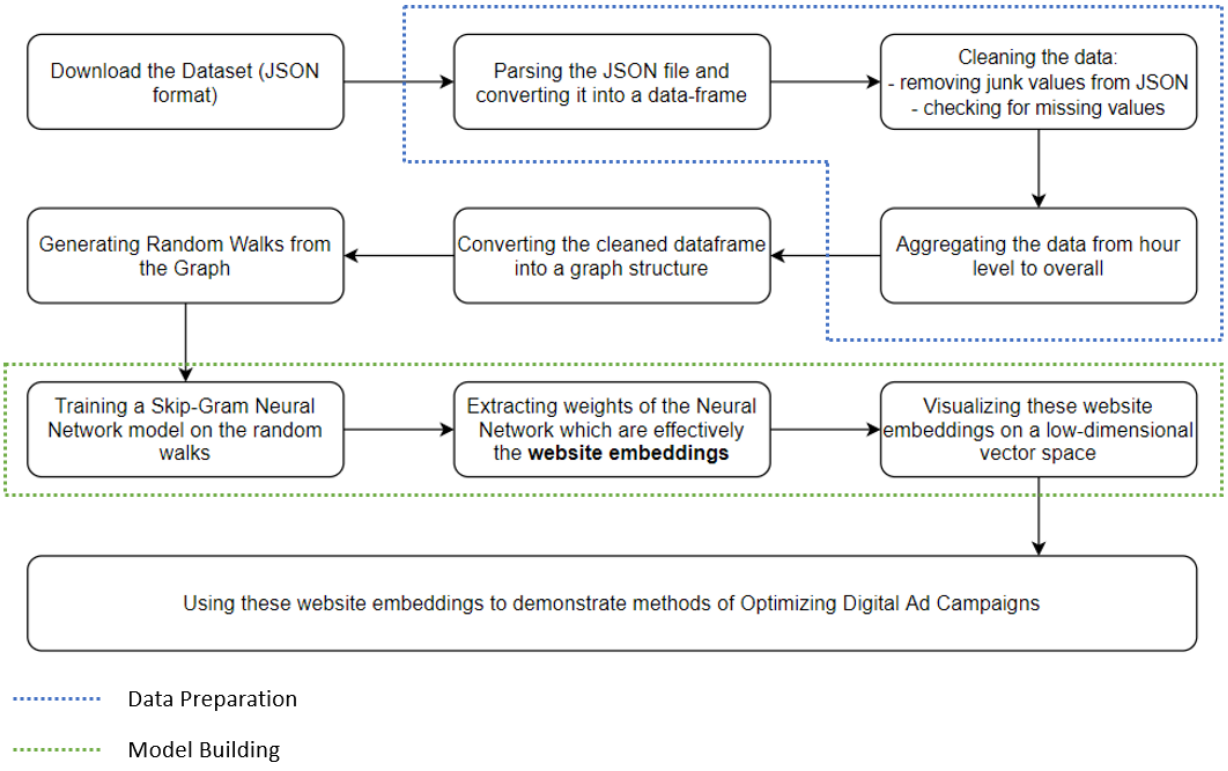


Figure 1: Methodology for generating website embeddings

Post generating the website embeddings, we'll investigate the applications of these embeddings in the optimization of digital campaigns. We explore three specific applications:

1. Predicting the next website that a user would visit based on their web history:
   The dataset for this is collected, with permission, from MiQ Digital India Pvt. Ltd, which is a company working in the marketing intelligence domain. The dataset contains 3 columns: timestamp, a hashed user identifier, and the website they visit. We will map the websites visited by the user to our embeddings to give us an understanding of where the user falls in our trained embedding vector space. Post that, we'll use a distance metric (Euclidian distance) to identify the sites that are closest to the user, which would be a data-driven indicator of where the user is likely to visit next.

2. Clustering the websites to get segments of similar websites
   Having generated the website embeddings and represented them in a vector space, we will be using the k-means clustering technique to identify the different clusters of websites.

3. Using the website embeddings as a predictor feature in classification problems
   One of the major machine learning applications in the digital advertising domain is the click prediction problem. The task here is to predict whether a user would click on an online ad based on their browsing behaviour. One of the predictor variables that features in this problem is the website visited by the user. In the conventional feature engineering process, these websites are usually one-hot encoded. However, we propose the use of our vector representations for these websites in place of one-hot encoding the feature, which would not only make the website categorical feature more intuitive but also make the model training more efficient as opposed to one-hot encoding.

7. Expected Outcomes

Post performing the described methodology, we expect to represent the websites in our dataset in form of low-dimensional vector representations, that we're referring to as embeddings.

We expect to use these website embeddings to optimize digital campaigns using the 3 applications discussed above:

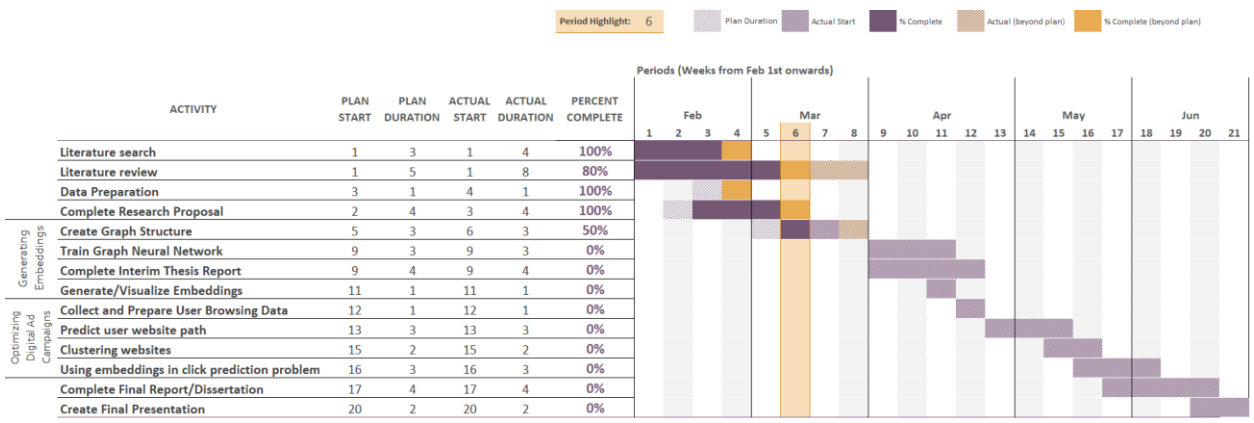
1. The embeddings should be able to predict the path of the user, which would allow a marketer to reach the user with the appropriate ad in the appropriate website
2. The embeddings should be able to generate segments of relevant websites which can be made available to a marketer to serve ads on
3. The embeddings should be capable of being used as predictor variables in prediction tasks within the digital advertising domain

8. Requirements / resources

Training the neural network with such a large training set would be computationally expensive. The most vital resource regarding this research would be additional GPU power, to handle the training computation power required. Along with this, we would be requiring additional cloud storage capacity where the embeddings and graph structures can be stored.

9. Research Plan

The Gantt chart below explains the tentative plan of action and milestones set for this research.



	ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
	Literature search	1	3	1	4	100%
	Literature review	1	5	1	8	80%
	Data Preparation	3	1	4	1	100%
	Complete Research Proposal	2	4	3	4	100%
Generating Embeddings	Create Graph Structure	5	3	6	3	50%
Generating Embeddings	Train Graph Neural Network	9	3	9	3	0%
Generating Embeddings	Complete Interim Thesis Report	9	4	9	4	0%
Generating Embeddings	Generate/Visualize Embeddings	11	1	11	1	0%
Optimizing Digital Ad Campaigns	Collect and Prepare User Browsing Data	12	1	12	1	0%
Optimizing Digital Ad Campaigns	Predict user website path	13	3	13	3	0%
Optimizing Digital Ad Campaigns	Clustering websites	15	2	15	2	0%
Optimizing Digital Ad Campaigns	Using embeddings in click prediction problem	16	3	16	3	0%
	Complete Final Report/Dissertation	17	4	17	4	0%
	Create Final Presentation	20	2	20	2	0%

Figure 2: Gannt Chart showing plan of action/milestones

References

Papers

Abiodun, O.I., Jantan, A., Omolara, A.E., Dada, K.V., Mohamed, N.A.E. & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*. [Online]. 4 (11). Available from: https://doi.org/10.1016/j.heliyon.2018.e00938.

Adolfsson, A., Ackerman, M. & Brownstein, N.C. (2019). To cluster, or not to cluster: An analysis of clusterability methods. *Pattern Recognition*. 88. p.pp. 13–26.

Agarwal, P. & Shukla, V.K. (2013). E-marketing Excellence: Planning and Optimizing Digital Marketing. *International Journals of Marketing and Technology*. 3 (March 2017). p.pp. 131–133.

Arthur, D. & Vassilvitskii, S. (2007). k-means++: The Advantages of Careful Seeding. *Proceeding of the 18th Annual ACM-SIAM Symposium of Discrete Analysis*. p.pp. 1027–1035.

Bala, M. & Deepak Verma, M. (2018). A Critical Review of Digital Marketing. *International Journal of Management*. 8 (10). p.pp. 321–339.

Berry, S. (2019). *What Is Contextual Targeting? (And Why Does It Even Matter?)*. [Online]. 2019. Available from: https://www.webfx.com/blog/marketing/contextual-targeting/.

Blum, A., Wardman, B., Solorio, T. & Warner, G. (2010). Lexical feature based phishing URL detection using online learning. *Proceedings of the ACM Conference on Computer and Communications Security*. p.pp. 54–60.

Bussche, D.A.F. von dem & Voigt, P. (2017). *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer International Publishing.

Choi, H., Mela, C.F., Balseiro, S. & Leary, A. (2019). Online Display Advertising Markets: A Literature Review and Future Directions. *SSRN Electronic Journal*.

Deshpande, I. (2019). *What Is Programmatic Advertising? Definition, Types, Channel, and Advantages*. [Online]. 2019. Available from: https://www.martechadvisor.com/articles/ads/what-is-programmatic-advertising/.

Enberg, J. (2019). *Global Digital Ad Spending 2019*. [Online]. 2019. Available from: www.emarketer.com/content/global-digital-ad-spending-2019.

García, J.J.L., Lizcano, D., Ramos, C.M.Q. & Matos, N. (2019). Digital marketing actions that

achieve a better attraction and loyalty of users: An analytical study. *Future Internet*. 11 (6). p.pp. 1–16.

Goldfarb, A. & Tucker, C. (2011). Online Display Advertising: Targeting and Obtrusiveness. *Marketing Science*. 30 (3). p.pp. 413–415.

Gonzalvez-Cabañas, J.C. & Mochón, F. (2016). Operating an Advertising Programmatic Buying Platform: A Case Study. *International Journal of Interactive Multimedia and Artificial Intelligence*. 3 (6). p.p. 6.

Gron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 1st Ed. O'Reilly Media, Inc.

Grover, A. & Leskovec, J. (2016). node2vec: Scalable feature learning for networks. *the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. p.pp. 855–864.

Hagberg, A.A., Schult, D.A. & Swart, P.J. (2008). Exploring network structure, dynamics, and function using NetworkX. *7th Python in Science Conference (SciPy 2008)*. (January 2008). p.pp. 11–15.

Hamilton, W.L., Ying, R. & Leskovec, J. (2017). Representation Learning on Graphs: Methods and Applications. *IEEE Data Eng. Bull.* [Online]. 40. p.pp. 52–74. Available from: http://arxiv.org/abs/1709.05584.

Hasan, M. & Zaki, M. (2011). A Survey of Link Prediction in Social Networks. *Social Network Data Analytics*. (March 2011). p.pp. 243–275.

Hofgesang, P.I. & Kowalczyk, W. (2006). Analysing clickstream data: From anomaly detection to visitor profiling. *Belgian/Netherlands Artificial Intelligence Conference*.

Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R. & Wu, A.Y. (2002). An Efficient k-Means Clustering Algorithm: Analysis and Implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 24. p.pp. 881–892.

Klein, S. (2010). *Introduction to Digital Advertising on Google, Facebook, & More*. [Online]. 2010. Available from: https://www.artonicweb.com/learn/digital-advertising/.

Le, H., Pham, Q., Sahoo, D. & Hoi, S.C.H. (2018). URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection. *ACM Conference, Washington, DC, USA*. [Online]. Available from: http://arxiv.org/abs/1802.03162.

Li, R.H., Yu, J.X., Qin, L., Mao, R. & Jin, T. (2015). On random walk based graph sampling.

Proceedings - International Conference on Data Engineering. 2015-May (April). p.pp. 927–938.

Li, X. & Mao, Y. (2012). *A survey on the generalized connectivity of graphs*. [Online]. (February 2014). Available from: http://arxiv.org/abs/1207.1838.

Lin, M.S., Chiu, C.Y., Lee, Y.J. & Pao, H.K. (2013). Malicious URL filtering - A big data application. *Proceedings - 2013 IEEE International Conference on Big Data, Big Data 2013*. p.pp. 589–596.

Maaten, L. van der & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*. [Online]. 1. p.pp. 1–48. Available from: http://www.cs.toronto.edu/~hinton/absps/tsne.pdf.

Marouchos, C. (2020). *Programmatic Advertising 101: Campaign Optimization*. [Online]. 2020. Available from: https://www.centro.net/blog/programmatic-101-campaign-optimization.

Meiss, M., Gonçalves, B., Ramasco, J.J., Flammini, A. & Menczer, F. (2010). Modeling Traffic on the Web Graph R. Kumar & D. Sivakumar (eds.). *Kumar R., Sivakumar D. (eds) Algorithms and Models for the Web-Graph. WAW 2010. Lecture Notes in Computer Science*. [Online]. 6516 (May 2014). Available from: http://link.springer.com/10.1007/978-3-642-18009-5.

Meiss, M.R., Menczer, F., Fortunato, S., Flammini, A. & Vespignani, A. (2008). Ranking web sites with real user traffic. *WSDM'08 - Proceedings of the 2008 International Conference on Web Search and Data Mining*. (October 2014). p.pp. 65–75.

Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013a). Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*. p.pp. 1–12.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. & Dean, J. (2013b). Distributed Representations of Words and Phrases and their Compositionality. *Advances in Neural Information Processing Systems*. (October).

Mishra, S., Sarkar, U., Taraphder, S., Datta, S., Swain, D., Saikhom, R., Panda, S. & Laishram, M. (2017). Multivariate Statistical Data Analysis- Principal Component Analysis (PCA). *International Journal of Livestock Research*. [Online]. (January). p.p. 1. Available from: http://www.ejmanager.com/fulltextpdf.php?mno=261590.

Murtagh, F. & Contreras, P. (2011). Methods of Hierarchical Clustering. *ArXiv*. abs/1105.0.

Murtagh, F. & Legendre, P. (2011). Ward's Hierarchical Clustering Method: Clustering Criterion and Agglomerative Algorithm. *ArXiv*. [Online]. (June). p.pp. 1–20. Available from: http://arxiv.org/abs/1111.6285%0Ahttp://dx.doi.org/10.1007/s00357-014-9161-z.

Naik, G.R. (2018). *Advances in Principal Component Analysis*. 1st Ed. Springer Singapore.

Perozzi, B., Al-Rfou, R. & Skiena, S. (2014). DeepWalk. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14*. [Online]. 2014, New York, New York, USA: ACM Press, pp. 701–710. Available from: http://dl.acm.org/citation.cfm?doid=2623330.2623732.

Ram, A., Jalal, S., Jalal, A.S. & Kumar, M. (2010). A Density Based Algorithm for Discovering Density Varied Clusters in Large Spatial Databases. *International Journal of Computer Applications*. 3 (6) p.pp. 1–4.

Sajjad, H.P., Docherty, A. & Tyshetskiy, Y. (2019). *Efficient Representation Learning Using Random Walks for Dynamic Graphs*. [Online]. Available from: http://arxiv.org/abs/1901.01346.

Sarveniazi, A. (2014). An Actual Survey of Dimensionality Reduction. *American Journal of Computational Mathematics*. 04 (02). p.pp. 55–72.

Savić, D. & Veinović, M. (2018). Challenges of General Data Protection Regulation (GDPR). In: *Sinteza 2018*. 2018.

Sharma, P. (2014). Centrality Measures in Social Networking: Study and Analysis Using NetDraw 2.138 in UCINET6. *International Journal of Engineering Research & Technology (IJERT)*. 3 (4). p.pp. 2583–2586.

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J. & Mei, Q. (2015). LINE: Large-scale Information Network Embedding. In: *Proceedings of the 24th International Conference on World Wide Web - WWW '15*. [Online]. 2015, New York, New York, USA: ACM Press, pp. 1067–1077. Available from: http://dl.acm.org/citation.cfm?doid=2736277.2741093.

Vathy-Fogarassy, Á. & Abonyi, J. (2013). *Graph-Based Clustering and Data Visualization Algorithms*. 1st Ed. Springer-Verlag London.

Wang, D., Cui, P. & Zhu, W. (2016). Structural Deep Network Embedding. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. [Online]. 13 August 2016, New York, NY, USA: ACM, pp. 1225–1234. Available

from: https://dl.acm.org/doi/10.1145/2939672.2939753.

Wang, F., Franco-Penya, H.-H., Kelleher, J.D., Pugh, J. & Ross, R. (2017). An Analysis of the Application of Simplified Silhouette to the Evaluation of k-means Clustering Validity. In: *13th International Conference on Machine Learning and Data Mining MLDM 2017*. 2017.

Yasmin, A., Tasneem, S. & Fatema, K. (2015). Effectiveness of Digital Marketing in the Challenging Age: An Empirical Study. *The International Journal of Management Science and Business Administration*. 1 (5). p.pp. 69–80.

Zhang, Z., Zhao, J. & Yan, X. (2018). A web page clustering method based on formal concept analysis. *Information (Switzerland)*. 9 (9).

Newspaper Articles, Online Articles and Websites

Enberg, Jasmine. "Global Digital Ad Spending 2019." *EMarketer*, Mar. 2019, www.emarketer.com/content/global-digital-ad-spending-2019

Berry, Sarah. "Contextual Targeting: What Is Contextual Advertising?" *WebFX Blog*, Dec. 2019, www.webfx.com/blog/marketing/contextual-targeting/.

Books

Han, J., Kamber, M., Pei, J. (2011). Data Mining: Concepts and Techniques, 3rd edition, *Morgan Kaufmann publications*, ISBN 978-0-12-381479-1