Utrecht University

Department of Information and Computing Science

Applied Data Science Master thesis

Predicting Negative Ties in Social Networks:

A Wikipedia Requests for Adminship case study

First examiner:
Javier Garcia-Bernardo

Second examiner:
Eva Jaspers

Candidate:
Sofoklis Repopoulos

In cooperation with:
Filip Chrzuszcz
Bo Staals

July 6, 2023

Acknowledgments

I would like to express my gratitude to both of my supervisors, Dr. Javier Garcia-Bernardo and Dr. Eva Jaspers, for their guidance and support throughout this research project. Additionally, I extend my acknowledgments to Elena Candellone for her critical corrections and genuine interest in the project. I am also thankful to my friends and colleagues Filip and Bo as their constant support has been a source of motivation during challenging times. Last but not least, I am thankful to my family and friends for their unwavering encouragement, understanding, and love.

Abstract

Social networks play a pivotal role in connecting individuals and fostering interactions in various domains. They serve as platforms for communication, information sharing, and community building. Predicting the nature of relationships, specifically negative ties, within social networks has garnered significant attention due to its potential impact on user experiences and network dynamics. This study focuses on the prediction of negative ties in social networks, specifically in the context of the labeled Wikipedia Requests for Adminship online social network. Three distinct models are employed to accomplish this task. Firstly, a Light Gradient Boosting Model (LGBM) utilizes graph topology attributes to make predictions. The LGBM leverages the structural characteristics of the network, such as node centrality and connectivity, to identify negative ties.

Secondly, a DistilBert language model is employed to process text data between users and their corresponding vote labels. The DistilBert model captures the semantic information embedded within the textual interactions, allowing for a more nuanced understanding of user sentiments and intentions. Finally, a Stacking Ensemble Model is employed to combine the predictions from the LGBM and DistilBert models. The Stacking Ensemble Model aggregates the predictions of the base models and employs a meta-learner to make the final predictions. Performance evaluation measures, including accuracy, precision, recall, F1-score, and elements of the confusion matrix, are used to assess the models' predictive capabilities. Presently, all models exhibit strong performance in detecting positive and negative signed links within the network. Notably, the DistilBert and Stacking Ensemble models consistently demonstrate superior performance across all classes. Future research should focus on addressing class distribution issues, incorporating diverse data, and exploring ensemble techniques to further enhance the predictive capabilities of these models.

Contents

1. Introduction

The focus of numerous social network studies has been on positive relational ties such as friendship or trust among individuals. However, the dark side of human interaction, where negative connections represent different forms of interpersonal conflict, intolerance or even abuse (Harrigan et al., 2020) is equally important and the extent to which positive and negative social network structure differs remain unclear. Given the serious problems associated with conflictual relationships, it remains without saying that conflictual interconnections routinely characterize social interaction (Maoz et al., 2007). Negative interactions can have a significant impact, since they can lead to the formation of cliques and communities that are less likely to interact with another, such as toxic communities with members actively engaging in bullying or harassment (Kaur & Singh, 2016).

1.1 Background

In many real-world social systems, relations between nodes can be represented as signed networks with positive and negative links. Online social networks such as Youtube, TikTok and Twitter are becoming popular among large number of people, as a source of forming virtual communities online. These communities are developed by creating profiles and maintaining personal contacts of each user through social interactions.

Heider in the 1940s studied the perception and attitude of individuals and introduced structural balance theory, which is an important social theory for signed networks. Cartwright and Harary (1956) further developed the theory and introduced the notion of balanced signed graph to characterize forbidden patterns in social networks. Signed network analysis has attracted much attention from multiple disciplines such as social psychology, physics and computer science and has evolved considerably from both data and problem-centric

perspectives. Signed networks observed in the physical world are often small but dense and clean. As a consequence, most early research about signed networks had mainly focused on developing theories to explain social phenomena in signed networks (Heider, 1946), (Cartwright & Harary, 1956).

Balance theory is naturally defined for undirected networks, whereas status theory (Guha et al., 2004; Leskovec et al., 2010a) is relevant for directed networks. Social status can be represented in a variety of ways, and it represents the prestige of nodes. In its most basic form, status theory suggests that user-node u_i has a higher status than u_j if there is a positive link from u_j to u_i or a negative link from u_i to u_j.

Sign prediction in social networks aims to forecast the positive or negative nature of relationships between users. It has implications for understanding social dynamics, information diffusion, and network analysis.The study by Leskovec et al., 2010b investigates signed networks in social media. They analyze the prevalence of positive and negative edges, studying the characteristics of signed edges, and examining the relationship between structural properties and the presence of positive or negative relationships. Their findings provide valuable insights into the dynamics of signed networks in online social platforms.

People's evaluations of one another are prevalent in all kinds of discourse, public and private, across ages, genders, cultures and social classes (Dunbar, 2004). Such opinions matter for establishing reputations and reinforcing social bonds. Research on signed social networks suggest that how one person will evaluate or link with another can often be predicted from the network they are embedded in. Linguistic sentiment analysis on the other hand suggests that one could leverage textual features to predict the valence of evaluative texts describing people. In some settings, textual data is sparse but the network structure is largely observed. In others, text is abundant but the network is partly or unreliably recorded. In either case, separate sentiment or signed-network models will miss or misread these signals. West et al., 2014 develop a graphical model that synthesizes network and linguistic information to make more and better predictions about both. Moving forward, this integrated approach holds great promise for uncovering deeper insights and enhancing prediction accuracy in

various domains where evaluations and social connections play a pivotal role.

1.2 Objective and research question

In the current work we study the social network structure of the Wikipedia Admin Requests network defined by votes for Wikipedia Adminship candidates, and particularly the task of predicting the negative ties of the network. Our main goal is to create a general classifier that combines attributes from the graphical properties of the network and linguistic information through text reviews that adminship candidates exchange with each other. Therefore, the main research question is "can we predict the negative ties in the Wikipedia Requests for Admin network utilizing part of or all available attributes?"

2. Data

2.1 Description of the data

For a Wikipedia editor to become an administrator, a request for adminship (RfA) must be submitted, either by the candidate or by another community member. Afterward, any Wikipedia member may cast a supporting, neutral, or opposing vote. This induces a directed, signed network where nodes represent Wikipedia members and edges represent votes. The dataset we work on contains all votes since the adoption of the RfA process in 2003 through May 2013. There is also a rich textual component in RfAs in the form of comments from users to candidates. Further information for the data along with downloadable files can be found in the official Stanford Network Analysis Project (SNAP) website: https://snap.stanford.edu/data/wiki-RfA.html.

Our analysis begins via exploring network-level metrics. Initially, we identify the number of users in the network and the number of edges-votes. We also explore the distribution of positive, negative and neutral signed edges. Due to the fact that some users ran for election several times throughout the available time period, we point out that the same voter/votee pair may contribute several votes, resulting in multiple edges between two users.

Metrics	
Nodes-users	11381
Edges-votes	198275
Negative Edges	39080 (19.70%)
Positive Edges	138247 (69.72%)
Neutral Edges	11676 (5.88%)

Table 2.1: Characteristics of the WikiRFA network. The percentages under each edge sign class represent the proportion of the corresponding signed edges compared to the total number of edges.

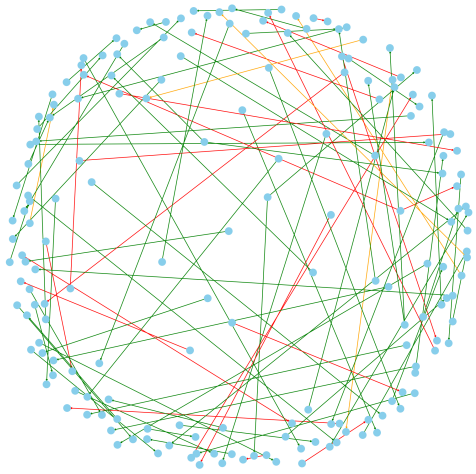

Figure 2.1: Random WikiRFA subgraph. Green, red and orange directed lines represent positive, negative and neutral votes respectively between users.

2.2 Preparation of the data

In order to prepare our dataset for further analysis we first account for duplicate pair values that come from users who ran for Adminship several times as mentioned in paragraph 2.1.1. Specifically, we keep the data entries that were marked with the most recent date, thus avoiding having multiple edges with the same direction for many pair of nodes. We also make sure to remove any missing votes from the dataset. The cleaned version of the network has the following characteristics:

Metrics	
Nodes	11381
Edges	189003
Negative Edges	38412 (20.32%)
Positive Edges	139362 (73.73%)
Neutral Edges	11229 (5.94%)

Table 2.2: Characteristics of the cleaned WikiRFA network. The percentages under each edge sign class represent the proportion of the corresponding signed edges compared to the total number of edges.

Snowball sampling

When dealing with large networks, it can be computationally expensive to perform operations on the entire graph. For example, calculating the cubed adjacency matrix A^3 of a network of n nodes would require $O(n^3)$ operations if a naive matrix multiplication algorithm is employed, or at best $O(n^{2.3728596})$ if the recent breakthrough algorithm of Alman and Williams, 2020 is used.

Snowball sampling (Goodman, 1961) is a suitable technique used to sample a subgraph of a large network very similar to the BFS sampling method. The algorithm can be summarized in the following steps:

Step	Description
1	Select a small number of seed nodes as the seed set
2	Collect the neighbors of each seed node
3	Add the neighbors to the subgraph and mark them as visited
4	Consider the neighbors of the newly added nodes and expand
5	Repeat steps 3 and 4 until the desired depth/number of layers

Table 2.3: Algorithmic steps of snowball sampling.

If we assume an average degree of d for the nodes of the initial graph and a desired depth of k, the time complexity of the Snowball sampling algorithm is $O(d^k)$. In contrast to random walks algorithms, the method manages to collect a representative number of nodes and edges that reflect the topology of the initial graph. However, we point out that the method may suffer from limitations such as bias towards highly connected regions and potentially miss isolated or low-degree nodes (Kurant et al., 2011).

We apply the Snowball sampling algorithm in the WikiRfa dataset with a random seed set of 10 nodes and the depth size set to 10. Comparing Table 2.2 with Table 4.3 we can to some extent conclude that the snowball sampling subgraph is representative of the initial graph as it maintains approximately the same proportion of positive, negative and neutral links.

3. Methods

3.1 Graph Topology Model

Link prediction is the problem of predicting the existence of a link between two nodes in a network (Liben-Nowell & Kleinberg, 2007). One of the types of traditional link prediction methods is heuristic methods and the first classifier we construct utilizes graph topology features based on heuristics and network topology features. The following are the ones we exploited.

3.1.1 Degree Distribution

In graph theory, the degree of a node is the number of edges that are incident to the node (Heider, 1946). In a directed signed network the degree of a node is decomposed in two parts: the in-degree and out-degree. The in-degree of a node is the number of incoming edges it has, and the out-degree the number of outgoing edges from that node. Analyzing the in and out degree of a network provides valuable information about how nodes receive and initiate connections with other nodes respectively.

According to Tang et al., 2014, the distributions of incoming and outgoing positive links for users usually follow heavy-tailed distributions, where a few users with large degrees are observed, while most users have small degrees. In a signed network, positive links are denser than negative links, with the negative links also having a heavy-tailed degree distribution where a few users have a large number of negative links, while most users have a few negative links. There seems to be an agreement between the previous statements and the in and out degree distributions of the processed WikiRFA subgraph network, even for the neutral class edges, judging by the histogram of in and out degrees depicted in Figure 3.1.

Nodes with higher positive indegree tend to exhibit a multitude of positive

relationships, thereby indicating a higher likelihood of positive edges. On the contrary, nodes with higher negative indegree often possess a substantial number of negative relationships, thus indicating a higher likelihood of negative edges. All these degree metrics can serve as useful predictors for discerning the sign edges.

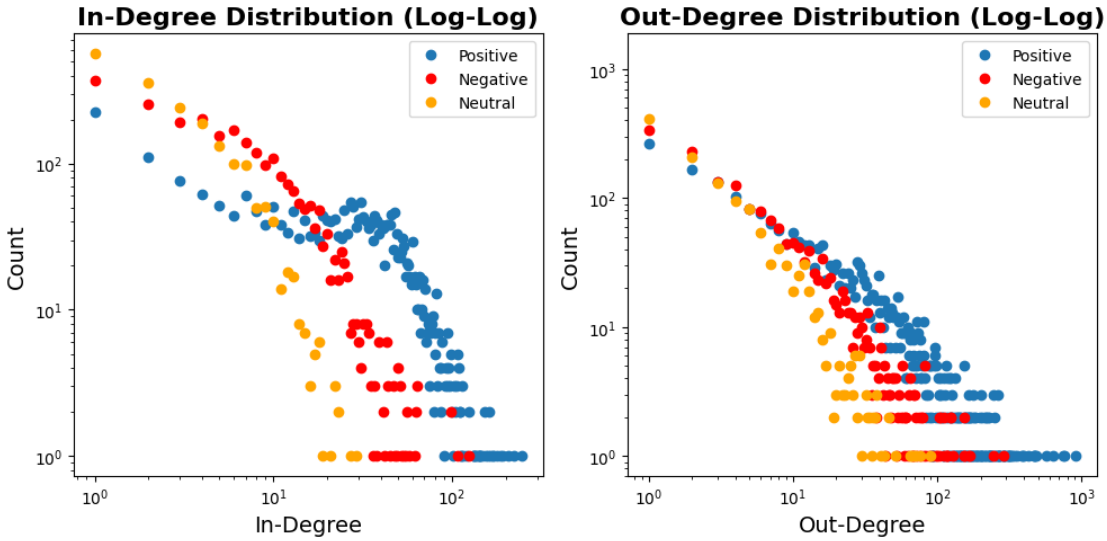

Figure 3.1: Log-log plots of In-Degree and Out-Degree Distribution of the Snowball subgraph.

3.1.2 Local Heuristics

Jaccard score measures the proportion of common neighbors two nodes have, meaning the proportion of neighbors two nodes (x, y) share as a measurement of their likelihood of having a link:

$$f_{\text{jaccard}}(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|},$$

where $\Gamma(x)$ is the neighborhood of node x, which means the set of nodes that are connected with x (Bass et al., 2013).

Also famous is the **preferential attachment** heuristic (Barabási & Albert, 1999), which uses the product of node degrees to measure the link likelihood. The basic assumption is that node x is more likely to connect to y if y has a high degree:

$$f_{\text{pa}}(x, y) = |\Gamma(x)| \cdot |\Gamma(y)|.$$

Resource allocation (RA) (Zhou et al., 2009) favors low-degree common neighbors using an aggressive down-weighting factor:

$$f_{\text{RA}}(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{|\Gamma(z)|}.$$

In many network setups, connections to people who are themselves influential will lend a person more influence than connections with less influential individuals. This effect can be quantified by the **eigenvector centrality** score denoted by EC_u for vertex u. We provide the formula below, where $\alpha_{u,v} = 1$ if vertex u is linked to vertex v and $\alpha_{u,v} = 0$ otherwise:

$$EC_u = \frac{1}{\lambda} \sum_{v \in V} \alpha_{u,v} EC_v.$$

If we define $\mathbf{x} = (x_1, x_2, \dots)$ to be the vector of centralities and assuming that we wish the centralities to be non-negative, it can be shown that λ is the largest eigenvalue of the adjacency matrix and $\mathbf{x}$ the corresponding eigenvector (Newman, 2018).

We also utilize the **Page Rank** (Page et al., 1999) algorithm which computes a ranking of the nodes in the graph based on the structure of the incoming links. In the Page Rank algorithm the importance of a node is higher if it is connected to other important nodes and a weight gets assigned. We can then identify negative and positive hubs and predict the edges of other nodes.

3.1.3 Clustering Coefficient

In graph theory clustering coefficient quantifies the degree to which nodes in a graph tend to cluster together. In social networks, nodes tend to create highly knit groups characterized by a relatively high density of ties (Watts & Strogatz, 1998). In our model we use the **local clustering coefficient** of a node, as it quantifies how close its neighbors are to forming a clique and thus potentially forming more positive or negative connections. The local clustering coefficient for a node is given by the proportion of the number of links between the vertices within the node's neighborhood, divided by the number of links that could possibly exist between them. For a directed graph with edges e_{ij} from the set E, N_i depicting the neighborhood of a node u_i and k_i the number of neighbors of vertex u_i, the formula for the local clustering coefficient is as follows:

$$C_i = \frac{|\{e_{jk} : u_j, u_k \in N_i, e_{jk} \in E\}|}{k_i(k_i - 1)}.$$

3.1.4 Adjacency matrix and structural balance theory

Structural balance theory is a generalization of Heider's social balance theory developed by Cartwright and Harary, 1956. The theory rests on the premise that certain configurations of positive and negative edges between individuals are socially more plausible than others. For example in the simple case of three individuals a, b, and c the left two configurations in figure 3.2 are more likely than the right two.

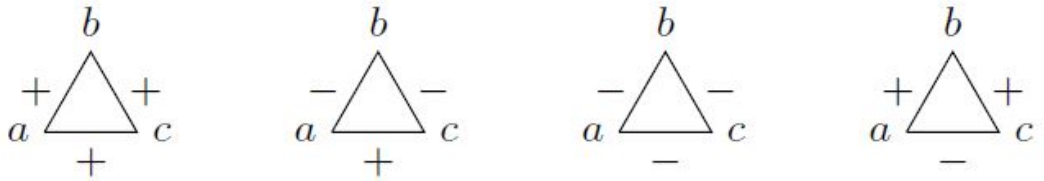

Figure 3.2: Balanced and Imbalanced Triangles (Chiang et al., 2011)

Triangles as the ones depicted in figure 3.2, are considered unbalanced if

they have an odd number of negative edges. A more intuitive reasoning from sociology is that “a friend of a friend is a friend" and “an enemy of an enemy is a friend". There is a connection between powers of the adjacency matrix A of a graph and the number of balanced or unbalanced cycles of length k, both for the directed and undirected case. Focusing on triangles (i.e. cycles of length 3), the (i, i) entry in the matrix A^3 represents the number of unbalanced triangles from node i and back to node i. Along with elements of the A^3 matrix, we utilize the number of unbalanced cycles of length 4 via the elements of the A^4 matrix and these are the predictors we choose to implement structural balance theory (Chiang et al., 2011). Calculating even higher-order powers of the adjacency matrix would require a lot of computational resources so we limit our approach to the calculation of the two previous matrices.

3.1.5 Status theory

Status theory explores the ways in which individuals and groups establish and maintain their social positions within a given society. Max Weber in his seminal work “The Theory of Social and Economic Organization” (Weber et al., 1948), examined the multidimensional nature of status, emphasizing its interplay with class and power in shaping social stratification. According to the work of Leskovec et al., 2010a, a positive directed link indicates that the creator of the link views the recipient as having higher status; and a negative directed link indicates that the recipient is viewed as having a lower status. In our study, we consider as a metric of status the positive in-degree of a node where nodes with more positive incoming edges have a higher status than those with less. Also the negative in-degree of nodes captures nodes that are “less-respected" inside the network. Both positive and negative in-degree of nodes are used as predictors in our graph model.

3.1.6 LGBM classifier

The Gradient Boosting Decision Tree (GBDT) is a widely used-machine learning algorithm that can achieve state of the art performances in many machine learning tasks such as multi-class classification (Li, 2012) and learning to rank

(Burges, 2010). In our multi-class classification problem, we use the Light Gradient Boosting Model (LGBM) by Ke et al., 2017, which is proven to speed up the training process of conventional GBDTs by up to over 20 times while achieving almost the same accuracy. GBDT is an ensemble model of decision trees, which are trained in sequence. In each iteration the GBDT learns the decision trees by fitting the negative gradients also known as the residual errors. The most time-consuming part in learning a decision tree is to find the best split points. A schematic figure of the process is provided below.

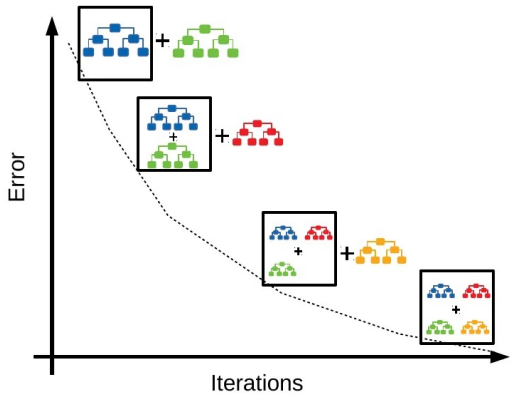

Figure 3.3: GBDT schema (Source: https://medium.com/swlh/gradient-boosting-trees-for-classification-a-beginners-guide-596b594a14ea)

In each iteration and for every feature, traditional engineering optimizations need to scan all the data instances to estimate the information gain of all possible split points, which is very time consuming. The LGBM algorithm manages to exclude a significant proportion of data instances with small gradients, and only use the rest to estimate the information gain using two novel techniques called *Gradient-based One-Side Sampling* and *Exclusive Feature Bundling*. Most decision tree learning algorithms grow trees by level (depth)-wise, whereas LGBM grows trees leaf-wise (best-first) based on the leaf with max delta loss (i.e. gradient) to grow. Leaf-wise algorithms tend to achieve lower loss than the level-wise algorithms. However, leaf-wise may cause over-fitting if the training dataset is too small, thus LGBM includes a `max_depth` parameter to limit tree growth (Figure 3.4, https://lightgbm.readthedocs.io/en/latest/Features.html#references).

Figure 3.4: Level-wise tree growth based learning vs LGBM leaf-wise tree growth learning.

3.2 DistilBert Model

An important data attribute that we cannot exploit in our graph topology model is the textual reviews between users. The last four years have seen the rise of large-scale pre-trained language models such as GPT-3 and BERT becoming a basic tool in many Natural Language Processing tasks (Devlin, 2018). In our problem, we want to utilize such a model to implement text classification based on the labeled positive negative or neutral sentiment each review has in our dataset. For the matter at hand, we deploy the DistilBert Model which is a smaller general-purpose language representation model of the BERT model, with outstanding performances on a wide range of tasks like its larger counterparts (Sanh et al., 2019).

BERT which stands for Bidirectional Encoder Representations from Transformers is a language representation model that is designed to pre-train bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. This results in the pre-trained BERT model that can be fine-tuned with just one additional output-layer to create state of the art models for a wide range of tasks, such as question answering and language inference, without complex task-specifc architecture modifications (Devlin, 2018). There are two steps in BERTs framework namely pre-training and fine-tuning. During pre-training the model is trained on unlabeled data over different pre-training tasks. For fine-tuning, the model is first initialized with the pre-trained parameters, and all of the parameters are fine-tuned using labeled data from the downstream tasks (e.g. question-answering task, text classification). BERT's model architecture is a multi-layer bidirectional Transformer encoder and its analysis is beyond the scope of our work. We refer the reader to Vaswani et al.,

2017 for the original implementation and further technical details.

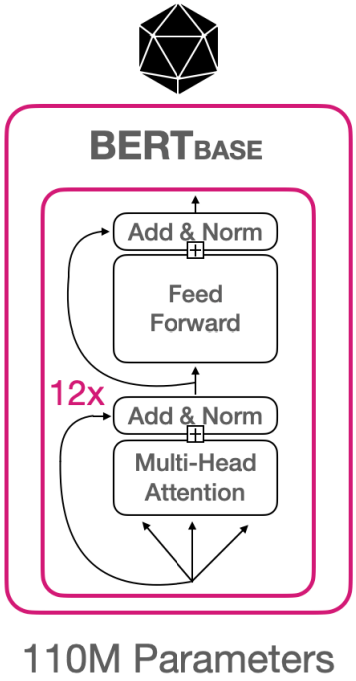

Figure 3.5: Architecture of BERTbase: 12 Transformer Blocks and 12 Attention Heads (i.e.the size of a Transformer Block), and a Hidden Size of 768 mathematical layers located between input and output that assign weights (to words) to produce a desired result.

The BERT$_{BASE}$ model has 110 million total parameters (Figure 3.5, https://huggingface.co/blog/bert-101) and scaling these types of models computational requirements has raised several environmental concerns (Schwartz et al., 2019; Strubell et al., 2020). The DistilBert model leverages knowledge distillation during pre-training phase and manages to retain 97% of BERTs language understanding capabilities while being 60% faster and 40% smaller. Knowledge distillation is a compression technique in which a compact model the "student" is trained to reproduce the behaviour of a larger model the "teacher" (Bucila et al. 2006, Hinton et al. 2015). A classification model is generally trained to predict an instance class by maximizing the estimated probability of gold labels. The DistilBert is trained with a distillation loss over the soft target probabilities (i.e. class probabilities in the output) of the BERT model. In the formula bellow t_i (resp. s_i) is a probability estimated by the teacher (resp. student):

$$L_{ce} = \sum_i t_i \cdot \log(s_i).$$

Following Hinton et al., 2015, the training loss is a linear combination of the *distillation loss* and the *masked language modelling loss*. Overall, DistilBert has about half the total number of parameters of BERT base and retains 95% of BERT's performances in language understanding.

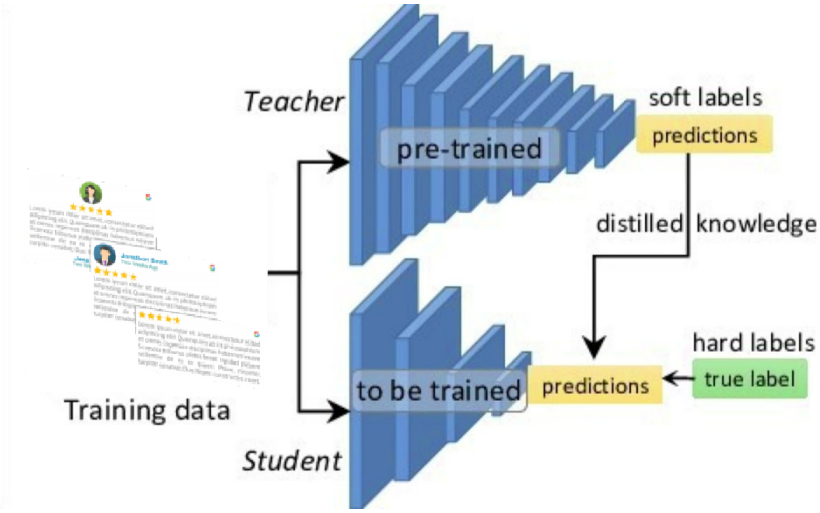

Figure 3.6: Knowledge distillation schema, https://pub.towardsai.net/a-gentle-introduction-to-knowledge-distillation-6240bf8eb8ea

3.3 Stacking Ensemble Model

Inspired from West et al., 2014 work, the last method we deploy is a stacking ensemble classifier modified to our needs and computational capabilities. An *ensemble* of classifiers combines the decision of individual classifiers in order to classify new instances. A main reason why one might want to deploy an ensemble of classifiers is that it may produce better results in case the training data cannot produce informative knowledge in order to select a single classifier (Dietterich, 2000). In our problem we have both textual and network topology data to work with and that is our motivation to try a joint model and specifically a Stacking Classifier. *Stacking* or *Stacked generalization* approach is based on the

production of a strong high-level learner with high generalized performance and to obtain this, a set of different classifiers are combined (Wolpert, 1992).

In order to deploy our stacking ensemble classifier we follow the steps bellow: First, we keep 10000 data entries from the total of 119256 from the Snowball sampled sub-data frame, to use as our final test set for all three models. Then, we split the remaining data to training set and test set and train the DistilBert and the LGBM classifiers on the training set. Following this, we make predictions from both models on the testing set and store the predicted values from the two models along with the actual values. Finally, we train the stacking ensemble model using a Random Forest classifier which takes as input the predicted values of the separate models on the test set and the actual values of the test set as the dependent variable. We note that normally cross-validation techniques or bootstrap sampling methods (Alexandropoulos et al., 2019) are used for the final step of the ensemble learner training, but due to limited computational resources, this was not possible to implement. Finally, we get predictions from all three methods for the final test set we created initially and compare their performance. We point out that the final test set was constructed so that it contains the same proportion of observations from each class as the snowball sampled graph, resulting in a representative data set.

4. Results

4.1 Model training

Initially, the LGBM and DistilBert models were fitted on a training set that was sampled without replacement using 90% of the observations in our snowball-sampled dataset. We stored the remaining observations as a test set, making sure to keep the same class imbalance through data stratification in both sets. The LGBM model was fine-tuned for the *learning rate, number of estimators, number of leaves* parameters using a Grid-Search algorithm (Belete & Huchaiah, 2021) that returned values 0.05, 500, and 50 respectively.

For the DistilBert language model, a basic text preprocessing was conducted to remove sentence terms that were inside double quotation marks because most of them expressed a rather obvious sentiment making the text classification trivial for any classifier. The language model was compiled with sparse categorical cross entropy loss function, an Adam optimizer, and a learning rate of 0.00005. DistilBerts training was restricted to 3 epochs, as the model overfits the training data quite fast after numerous experimentation attempts. A callback parameter was set in model compiling so that the best-performing weights are returned after fitting the model. Plots of model loss and accuracy training can be found in table 4.3 (Appendix A). As mentioned in paragraph 3.3, the Stacking Ensemble model is fitted using the predictions made by LGBM and DistilBert models on the test set as an input to a Random Forest Classifier with the number of estimators (i.e trees) equal to 100. The three models are evaluated on the final test set containing 10000 observations with the same class imbalance as the initial snowball sampled dataset. Source code that reproduces plots and outputs of methods can be found through this link: **https://github.com/SofoRep/ADS-Thesis-Project**. Code is shared under MIT licence.

4.2 Results

To evaluate model performance, we studied the elements of the confusion matrix for each model and calculated Precision, Recall, F1-score and Accuracy metrics. More information about the metrics can be found on Appendices B and C. Starting with the confusion matrices bellow (Figure 4.1), rows represent the actual values of each class and columns the predicted values respectively. We can divide the diagonal elements with the row (or column) sum of the corresponding matrix entries to get the Recall (resp. Precision) of each class. For example, the Recall of the LGBM model of the negative class is $\frac{1376}{1376+229+306} = 0.72$ and Precision for that class is $\frac{1376}{1376+205+792} = 0.58$.

Figure 4.1: Confusion Matrices of all models.

We also provide the models accuracy where DistilBert and Stacking Ensemble models outperform the LGBM model with almost identical scores:

Model	Accuracy
LGBM	0.7865
DistilBert	0.9068
Stacking Ensemble model	0.8945

Table 4.1: Model Accuracy

In table 4.2 we present summarized Precision, Recall and F1-scores for all the models and classes. The same results in the form of barplots can be found in Appendix 4.4. In terms of **Precision** of the positive class, DistilBert and Stacking Ensemble classifiers have the highest and identical score (0.95). For Precision of the neutral class, Distilbert model scores the highest (0.67) followed by LGBM and Stacking Ensemble model, and for the negative class Stacking Ensemble model has the highest score (0.80) followed by DistilBert and LGBM classifiers. Similarly, DistilBert and Stacking Ensemble model have the highest and identical **Recall** score of 0.97 for the Positive class. For the neutral Class Stacking ensemble model has the highest Recall score (0.51) and for the negative class DistilBert (0.84).

Model	**Precision**		
	Positive	**Neutral**	**Negative**
LGBM	0.92	0.23	0.58
DistilBert	**0.95**	**0.67**	0.79
Stacking Ensemble	**0.95**	0.48	**0.80**
Model	**Recall**		
	Positive	**Neutral**	**Negative**
LGBM	0.84	0.31	0.72
DistilBert	**0.97**	0.38	**0.84**
Stacking Ensemble	**0.97**	**0.51**	0.73
Model	**F1-score**		
	Positive	**Neutral**	**Negative**
LGBM	0.88	0.27	0.64
DistilBert	**0.96**	**0.49**	**0.81**
Stacking Ensemble	**0.96**	**0.49**	0.77

Table 4.2: Precision, Recall and F1-scores for all models.

Last but not least, DistilBert and Stacking Ensemble models outperform the LGBM model in terms of **F1 score** with identical values for positive and neutral class (0.96 and 0.49 respectively). The model with the highest F1-score for the negative class is DistilBert (0.81). An additional famous performance metric called Area Under the Curve (Bradley, 1997) that comes to terms with these statements is presented in Figure 4.5 (Appendix C).

Topology Model Feature Importance

LGBM topology model utilizes a variety of graph attribute features that we can rank after we fit the model to the training set, and examine their contribution to the algorithm's performance. We calculate feature importance based on the "split" method of the LGBM model, which measures the number of times a feature is used to split the data across all the trees in the ensemble (Scornet, 2020). Features with higher importance values have contributed more to the reduction in impurity during the tree-building process. They have been more frequently used for splitting the data and have had a greater impact on the model's decision-making.

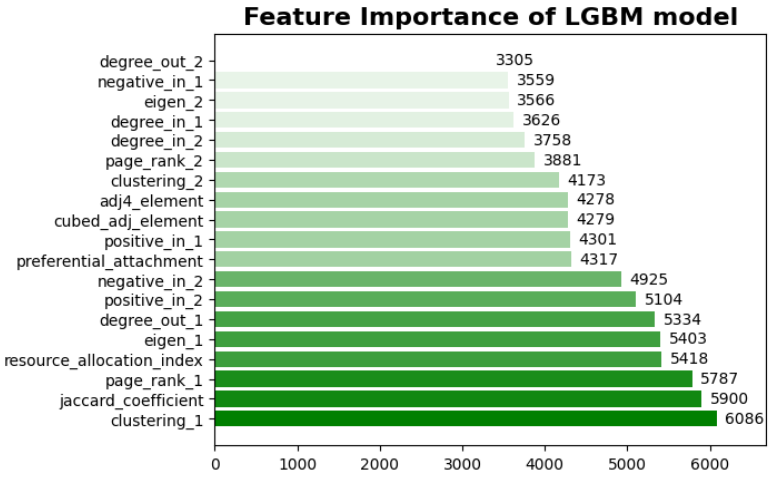

Figure 4.2: LGBM feature importance. Attributes with higher scores are considered more important.

Looking at the feature importance values of figure 4.2 we observe that the clustering coefficient attribute of starting nodes of each edge (`clustering_1`) has the highest importance value of 6086. This could mean that the community that the voter belongs to plays a significant role in the attitude expressed to the other user. On the other hand, the total out-degree (`degree_out2`) of the target-user (i.e. end of each edge) has an importance value of 3305, suggesting it has a relatively lower impact on the model's predictions compared to `clustering_1`. Notably so, status theory which is implemented through the positive and negative incoming edges of a node, seems to affect mostly the target nodes-users as their importance is higher than the starting nodes-users. Structural balance theory which is implemented through the elements of the third (`cubed_adj_`-

`element`) and fourth (`adj4_element`) power of the adjacency matrix A, is also significant with corresponding values 4279 and 4278. Similar conclusions can be made for the rest of the graph attributes.

4.3 Overview of the results

Overall, Precision, Recall and F1 score metrics are much higher for the positive and negative classes in all three models. This can indicate that it is easier for the models to distinguish the positive and negative signed links between two users in the WikiRfa social network. To answer our research question, the model that efficiently manages to identify most of the negative ties in the WikiRfa network is the DistilBert Model, followed by the Stacking Ensemble model and LGBM. This statement is based on all used evaluation metrics and specifically Recall of the Negative Class which expresses the proportion of negative edges that models managed to find out of all true negative links. What is more, DistilBert and Stacking Ensemble models seem to perform the best on all edge sign classes predictions with very similar scores on all used performance metrics.

Discussion

All models struggle to correctly identify neutral signed edges compared to positive and negative signed links in the network. However, working with such an imbalanced dataset, where approximately only 7% of the observations belong to the neutral and 30% to the negative class, justifies to a degree their weakness. It would be interesting for future work to augment the data with more observations of the neutral and negative classes and utilize a method to add "neutral" and "negative" sentiment text along with each new edge accordingly. Balancing the dataset might lead to better results primarily in the neutral sign class.

Stacking Ensemble model performs better than the LGBM model in all calculated performance metrics, with very similar results as the DistilBert. Stacking classifiers in general try to combine the strengths and weaknesses of "weaker" learners in order to increase the overall performance (Alexandropoulos et al., 2019). In our case however, DistilBert model outperforms the LGBM model in all aspects, and in the Stacking classifiers effort to combine the strengths of

two different worlds (Gradient Boosting Classifier and a Transformer Language Model), more weight is given to DistilBert model. Further research could make the LGBM model competitive against a Language model such as DistilBert and therefore lead to an improved Stacking classifier.

4.4 Conclusion

In this study, three separate models were utilized in order the predict the edge signs of the labeled Wikipedia Requests for Adminship online social network. Initially, a representative subgraph was sampled using the Snowball graph sampling technique that made the calculations feasible considering the project's timeframe. The first model, a Light Gradient Boosting Model, was assembled with graph topology attributes solely and fine-tuned to an extent with a Cross-Validation Grid Search Algorithm. Next, the text data-reviews between users along with the corresponding vote labels were fed as input to a DistilBert language model. Finally, an effort was made to combine the LGBM and the DistilBert models through a Stacking Ensemble Model to make better predictions.

The performance of the models was assessed using accuracy, precision, recall, F1-score, and elements of the confusion matrix on the final test set. Overall, all models performed well in detecting the positive and negative signed links of the network with high scores in almost all metrics. Characteristically, accuracy was above 78% for all models and the F1-scores for positive and negative were above 80% for the DistilBert and Stacking Ensemble Model. In the end, the DistilBert and Stacking Ensemble models proved to be the most consistent within all classes.

There are several improvements that could increase model performance, such as a better class distribution of the available data and incorporating data of different study areas, especially for the graph topology model. Moreover, a similar Ensemble approach may lead to better results combined with a well-balanced dataset. Further research can look into the application of these enhancements and possibly contribute to even more outstanding predictions.

Appendix A

Metrics	
Nodes	3452
Edges	119256
Negative Edges	22804 (19.12%)
Positive Edges	89225 (74.81%)
Neutral Edges	7227 (6.06%)

Table 4.3: Characteristics of the Snowball sampled subgraph of the WikiRFA network.

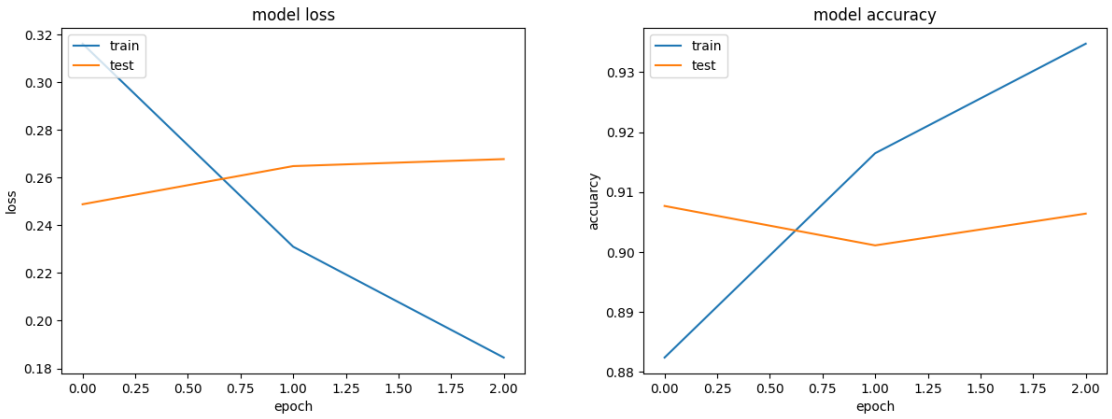

Figure 4.3: Distilbert accuracy and loss history during training.

Appendix B

Confusion matrix

The confusion matrix is a cross table that records the number of occurrences between two raters, the true classification and the predicted classification. The classes are listed in the same order in the rows as in the columns, therefore the correctly classified elements are located on the main diagonal from top left to bottom right and they correspond to the number of times the two raters agree (Grandini et al., 2020).

Accuracy

Accuracy is a metric that generally describes how the model performs across all classes. It is meaningful to use this metric when all classes are of equal importance. Formally, accuracy is calculated as the ratio between the number of correct predictions to the total number of predictions (Grandini et al., 2020). In the accuracy formula provided below, tp and tn represent the true positive and true negative elements which are the elements correctly classified by the model and they are on the main diagonal of the confusion matrix. fp (respectively fn) are the elements that have been labeled as positive (negative) by the model but are actually negative (respectively positive):

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn}.$$

Accuracy tends to hide strong classification errors for classes with a few units, and thus, it is not possible to identify the classes where the model is working worse. Accuracy is a reliable performance metric if the testing dataset is quite balanced, meaning that the classes are almost the same size.

Appendix C

Precision

Precision is the fraction of True Positive elements divided by the total number of positively predicted units (column sum of the predicted positives in the confusion matrix). Precision can also be calculated for each class where tp_k are the correctly classified units for the class, whereas fp_k are the wrongly classified elements on the corresponding column class of the confusion matrix:

$$\text{Precision} = \frac{tp}{tp + fp}, \qquad \text{Precision}_k = \frac{tp_k}{tp_k + fp_k}.$$

Recall

Recall is the fraction of True Positive elements divided by the total number of positively classified units (row sum of the actual positives in the confusion matrix). The Recall measures the model's predictive accuracy for the positive class and intuitively measures the ability of the model to find all the Positive units in the dataset. Extending recall in each individual class (fn_k are the wrongly classified elements on the row class of the confusion matrix) leads to the second provided formula:

$$\text{Recall} = \frac{tp}{tp + fn}, \qquad \text{Recall}_k = \frac{tp_k}{tp_k + fn_k}.$$

F1 score

The F1 score is a weighted average between Precision and Recall, with F1 score reaching its best value at 1 and worst score at 0. F1-score is also extended to each individual class as shown below:

$$\text{F1\_score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \qquad \text{F1\_score}_k = \frac{2 \cdot \text{Precision}_k \cdot \text{Recall}_k}{\text{Precision}_k + \text{Recall}_k}.$$

Appendix D

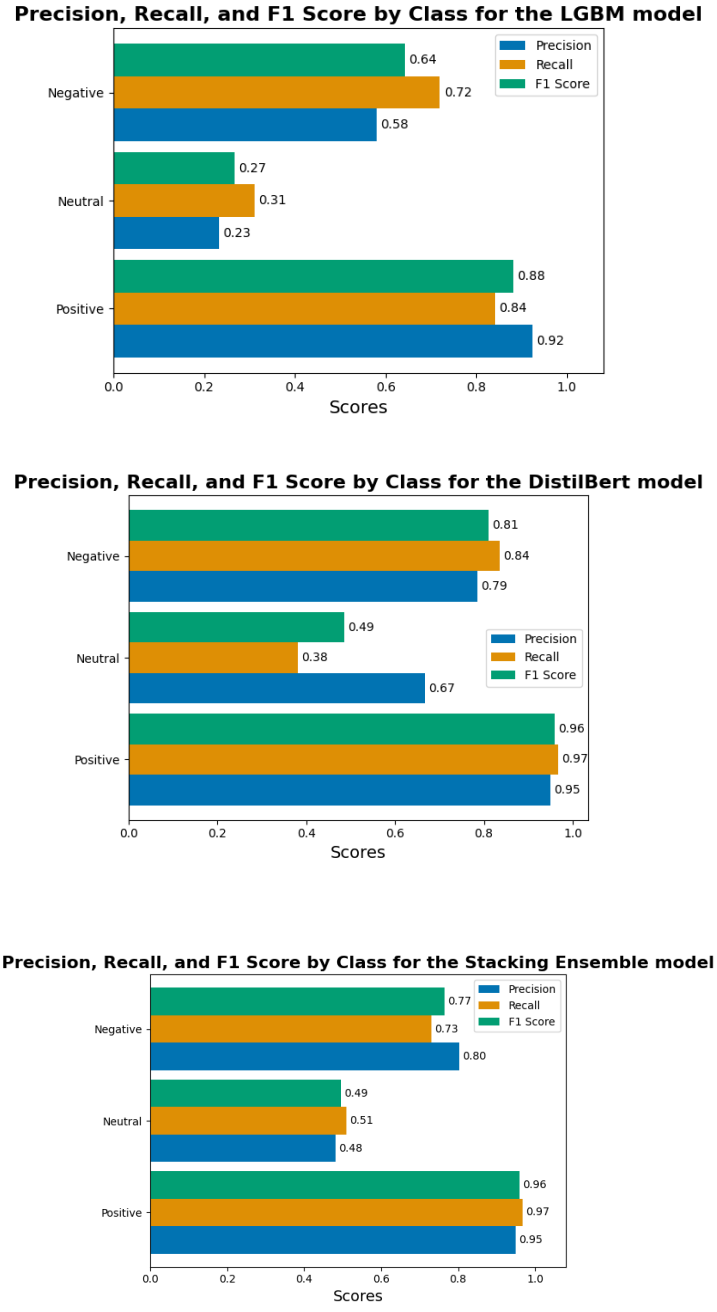

Figure 4.4: Barplots of Precision, Recall and F1-score metrics of models for all classes.

Appendix E

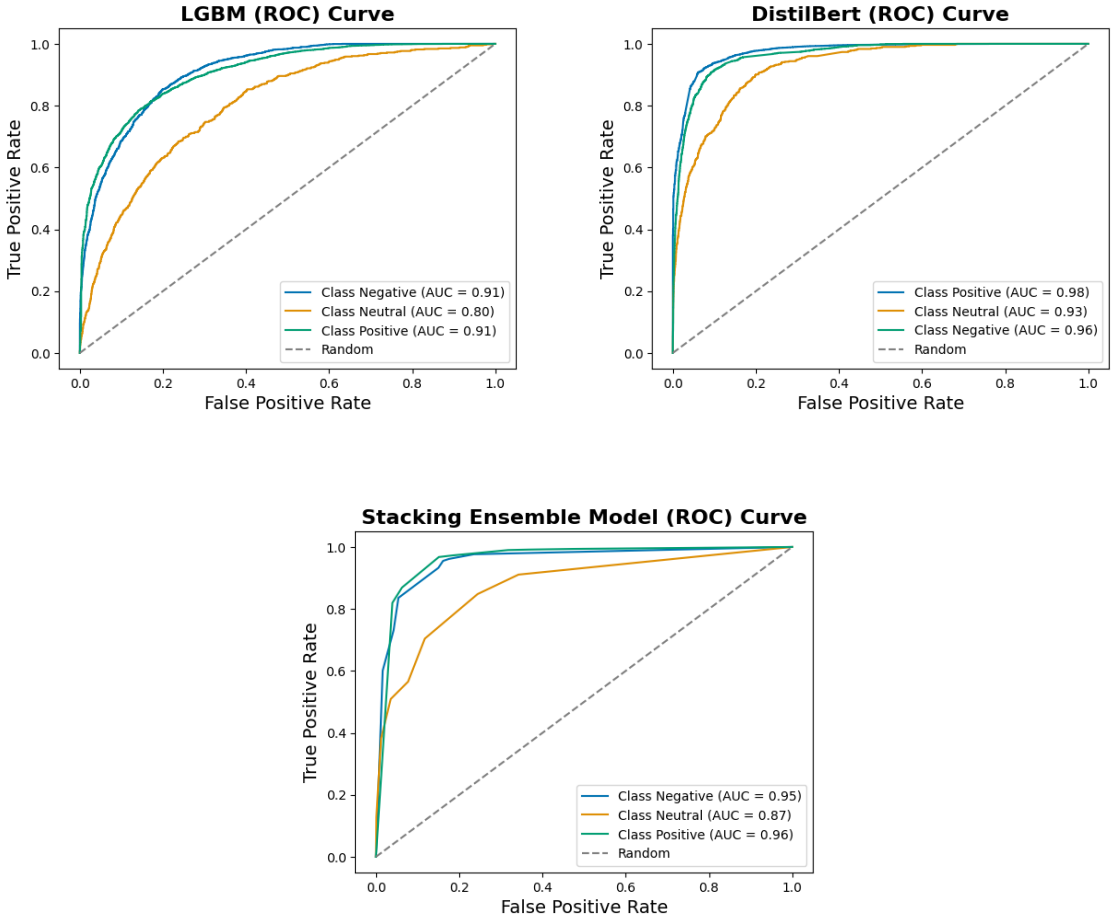

Figure 4.5: ROC curves for all models and classes.

The Receiver Operating Characteristic (ROC) curve is a good way of visualizing a classifier's performance in order to select a suitable operating point, or decision threshold (Bradley, 1997). The Curve is created by calculating True Positive ($\frac{tp}{tp+fn}$) and False Positive Ratio ($\frac{fp}{tn+fp}$) for every possible classification threshold. The Area Under the ROC Curve, (AUC) represents the degree or separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC value for each class, the better the model at predicting the class.

Bibliography

Harrigan, N., Labianca, G., & Agneessens, F. (2020). Negative ties and signed graphs research: Stimulating research on dissociative forces in social networks. *Social Networks*, *60*, 1–10. https://doi.org/10.1016/j.socnet.2019.09.004

Maoz, Z., Terris, L. G., Kuperman, R. D., & Talmud, I. (2007). What is the enemy of my enemy? causes and consequences of imbalanced international relations, 1816–2001. *The Journal of Politics*, *69*(1), 100–115. https://doi.org/10.1111/j.1468-2508.2007.00497.x

Kaur, M., & Singh, S. (2016). Analyzing negative ties in social networks: A survey. *Egyptian Informatics Journal*, *17*(1), 21–43. https://doi.org/10.1016/j.eij.2015.08.002

Heider, F. (1946). Attitudes and cognitive organization. *The Journal of Psychology*, *21*(1), 107–112. https://doi.org/10.1080/00223980.1946.9917275

Cartwright, D., & Harary, F. (1956). Structural balance: A generalization of heider's theory. *Psychological Review*, *63*(5), 277–293. https://doi.org/10.1037/h0046049

Guha, R., Kumar, R., Raghavan, P., & Tomkins, A. (2004). Propagation of trust and distrust. https://doi.org/10.1145/988672.988727

Leskovec, J., Huttenlocher, D. P., & Kleinberg, J. (2010a). Predicting positive and negative links in online social networks. https://doi.org/10.1145/1772690.1772756

Leskovec, J., Huttenlocher, D. P., & Kleinberg, J. (2010b). Signed networks in social media. https://doi.org/10.1145/1753326.1753532

Dunbar, R. I. M. (2004). Gossip in evolutionary perspective. *Review of General Psychology*, *8*(2), 100–110. https://doi.org/10.1037/1089-2680.8.2.100

West, R., Paskov, H. S., Leskovec, J., & Potts, C. (2014). Exploiting social network structure for person-to-person sentiment analysis. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1409.2450

Alman, J., & Williams, V. V. (2020). A refined laser method and faster matrix multiplication. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2010.05846

Goodman, L. A. (1961). Snowball sampling. *Annals of Mathematical Statistics*, *32*(1), 148–170. https://doi.org/10.1214/aoms/1177705148

Kurant, M., Markopoulou, A., & Thiran, P. (2011). Towards unbiased bfs sampling. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1102.4599

Liben-Nowell, D., & Kleinberg, J. (2007). The link-prediction problem for social networks. *Journal of the Association for Information Science and Technology*, *58*(7), 1019–1031. https://doi.org/10.1002/asi.20591

Tang, J., Hu, X., & Liu, H. (2014). Is distrust the negation of trust? https://doi.org/10.1145/2631775.2631793

Bass, J. I. F., Diallo, A., Nelson, J., Soto, J., Myers, C. L., & Walhout, A. J. (2013). Using networks to measure similarity between genes: Association index selection. *Nature Methods*, *10*(12), 1169–1176. https://doi.org/10.1038/nmeth.2728

Barabási, A., & Albert, R. (1999). Emergence of scaling in random networks. *Science*, *286*(5439), 509–512. https://doi.org/10.1126/science.286.5439.509

Zhou, T., Lü, L., & Zhang, Y. (2009). Predicting missing links via local information. *The European Physical Journal B*, *71*(4), 623–630. https://doi.org/10.1140/epjb/e2009-00335-8

Newman, M. F. (2018, October 18). *Mathematics of networks*. https://doi.org/10.1093/oso/9780198805090.003.0006

Page, L. M., Brin, S., Motwani, R., & Winograd, T. (1999). The pagerank citation ranking : Bringing order to the web. *98*, 161–172. http://dbpubs.stanford.edu:8090/pub/1999-66/

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, *393*(6684), 440–442. https://doi.org/10.1038/30918

Chiang, K., Natarajan, N., Tewari, A., & Dhillon, I. S. (2011). Exploiting longer cycles for link prediction in signed networks. https://doi.org/10.1145/2063576.2063742

Weber, M., Henderson, A. H., & Parsons, T. (1948). Max weber: The theory of social and economic organization. *Yale Law Journal*, *57*(4), 676. https://doi.org/10.2307/793128

Li, P. (2012). Robust logitboost and adaptive base class (abc) logitboost. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1203.3491

Burges, C. J. C. (2010). From ranknet to lambdarank to lambdamart: An overview. *Microsoft Research Technical Report MSR-TR-2010-82*. https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/MSR-TR-2010-82.pdf

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *30*, 3149–3157. https://papers.nips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

Devlin, J. (2018, October 11). *Bert: Pre-training of deep bidirectional transformers for language understanding*. http://arxiv.org/abs/1810.04805

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *30*, 5998–6008. https://arxiv.org/pdf/1706.03762v5

Schwartz, R., Dodge, J., Smith, N. A., & Etzioni, O. (2019). Green ai. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1907.10597

Strubell, E., Ganesh, A., & McCallum, A. (2020). Energy and policy considerations for modern deep learning research. *Proceedings of the ... AAAI Con-*

ference on Artificial Intelligence, *34*(09), 13693–13696. https://doi.org/10.1609/aaai.v34i09.7123

Hinton, G., Vinyals, O., & Dean, J. (2015). *Distilling the knowledge in a neural network*. arXiv: 1503.02531 [stat.ML].

Dietterich, T. G. (2000, January 1). *Ensemble methods in machine learning*. Springer Science+Business Media. https://doi.org/10.1007/3-540-45014-9_1

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, *5*(2), 241–259. https://doi.org/10.1016/s0893-6080(05)80023-1

Alexandropoulos, S. N., Aridas, C. K., Kotsiantis, S., & Vrahatis, M. N. (2019, January 1). *Stacking strong ensembles of classifiers*. Springer Science+Business Media. https://doi.org/10.1007/978-3-030-19823-7_46

Belete, D. M., & Huchaiah, M. D. (2021). Grid search in hyperparameter optimization of machine learning models for prediction of hiv/aids test results. *International Journal of Computers and Applications*, *44*(9), 875–886. https://doi.org/10.1080/1206212x.2021.1974663

Bradley, A. P. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, *30*(7), 1145–1159. https://doi.org/10.1016/s0031-3203(96)00142-2

Scornet, E. (2020). Trees, forests, and impurity-based variable importance. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2001.04295

Grandini, M., Bagli, E., & Visani, G. (2020). Metrics for Multi-Class Classification: an Overview. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2008.05756