THE THIRD ANNUAL

GREAT LAKES DATA SCIENCE SYMPOSIUM

FRIDAY, MAY 1, 2020
2 TO 6:10 P.M.

VIRTUAL EVENT ON MICROSOFT TEAMS

GO TO ***MERCYHURST.EDU/SYMPOSIUM*** TO JOIN.

MERCYHURST UNIVERSITY

THE THIRD ANNUAL

GREAT LAKES DATA SCIENCE SYMPOSIUM

Program Chair & Proceedings Editor: M. Afzal Upal, PhD
Chair of Computing & Information Science Department
Mercyhurst University
501 E 38th St
Erie, PA, 16546

Chair's Message

Welcome to the proceedings of the Third Annual Data Science Symposium. Even through this year's symposium was held virtually due to the Covid-19 pandemic, the quality of the works remains the same as the last two years. This year's proceedings include 13 papers that have been divided into the following five groups:

- Group 1 contains papers that use data science to predict some aspect of human behavior.
  - Winschel uses it to predict scoring behavior of NBA players.
  - Mahmud uses machine learning to predict American voter's party affiliations.
  - Zacherl and Bohara uses machine learning to predict student behavior.

- Group 2 contains papers that analyze social media messages.
  - Bahntge analyzes Twitter messages to understand differences between regional uses of the English language.
  - Mincewicz analyzes Twitter messages to understand level of violence in demonstrations.
- Group 3 contains papers that use deep learning.
  - Sim uses the deep learning model MelGAN-VC to mimic the voice cloning
  - Decker uses the masked R-CNN to predicting the position of keypoints on an electric railroad pantograph.
  - Vu employs a deep learning model, among others, to distinguish between illegal money laundering transactions and ordinary financial transactions.
- Group 4 contains papers that address mental health issues
  - Chuhan analyzes relationship between mental health and employment status.
  - Gonzalez studies factors that predict whether someone is likely to seek mental health treatment or not.
- The final paper develops a machine learning model to distinguish benign software from malware.

Table of Contents

Daily Fantasy Basketball Lineup Optimizer

Austin Winschel
Department of Computing &
Information Science
Mercyhurst University, Erie, PA
awinsc08@lakers.mercyhurst.edu

Abstract

Sports analytics has been on the rise and fantasy sports are more popular than ever. Machine learning and data collection for sports are only improving. This paper looks at daily fantasy sports, specifically basketball. The goal is to predict a players performance in terms of fantasy points in any given game. This was converted to a binary classification problem by looking at whether a player will score above a certain threshold in any given game. The data used consists of every game played by every active player in the NBA. Logistic regression, random forests, and an ensemble method are used, and the results are compared. The results are modestly promising.

Keywords

Daily Fantasy Sports, DFS, NBA, Sports Analytics, Logistic Regression, Random Forest, Support Vector Machine, K-Nearest-Neighbors, Ensemble.

1. Introduction

Fantasy sports and sports gambling have been increasing in popularity over the past decade. According to the Fantasy Sports Trade Association, fantasy sports participation rose from 32 million players in 2010 to 59.3 million in 2017 or an 85.3% increase. On average, fantasy sports players spend $566 on fantasy sports related costs over a 12 month period [5]. The number of ways that you play fantasy sports has increased as well. The different ways to play fantasy sports can be broken down into season long leagues and daily fantasy competitions, also known as daily fantasy sports (DFS). Season long leagues are the traditional way to play fantasy sports, where you draft a team and face off against the other players (usually 8-12). Your record is recorded for playoff seeding and the winner of the playoffs wins the league. The lineup constraint here is that each player can only be on one team at a time. DFS contests work differently, they last for a single set of games, within a given day. Winners are selected differently based on the type contest you enter, but each contest has a set number of players that receive a payout. Your fantasy score has to be in that top number of players to win. The lineup constraint in DFS is also different. It contains a position constraint and a salary constraint which will be detailed below.

DFS has been popularized in the past decade with companies such as FanDuel and DraftKings (also referred to as DK). We will be focusing on DK here, and specifically with NBA basketball. A little background knowledge is needed to understand how the lineups are constructed. There are five positions in basketball, which are point guard (PG), shooting guard (SG), small forward (SF), power forward (PF), and center (C). A DK lineup consists of eight players and must include players from at least two different NBA games. The eight roster positions are PG, SG, SF, PF, C, G, F, and Util. The G stands for guard and a PG or SG can be placed in that position. The F stands for forward and a SF or PF can be placed in that position. Util stands for utility and any player can be placed in that position. Along with the position constraint there is also a salary constraint. Players are assigned a salary value every day and there is a salary cap of $50,000 [4].

With the aforementioned increase in participation in fantasy sports and the progression of sports analytics, it is only natural that these two fields would connect. Sports analytics gained national attention from the release of the 2011 movie *Moneyball* [9], based on the book *Moneyball: The Art of Winning an Unfair Game*. This book details the journey of Billy Beane and the Oakland Athletics as they used sabermetrics, or empirical based analysis in baseball, to improve their player evaluation [6]. The movie brought the book more attention and brought to light the pushback received when attempting to integrate this type of analysis to sports. Decisions were typically made by gut feelings or past traditions [1]. Since it has taken some time for analytics to be accepted in sports, research on the topic is all relatively recent. Research in the fantasy aspect of sports is even more recent.

DK officially launched in 2012 [4]. Because it is a new way of playing fantasy sports and the fact that these algorithms are usually proprietary, little research has been published on the topic. Here, the goal was to classify a players fantasy points on a given night into a specific range.

2. Relevant Work

Sports analytics has continued to grow in popularity, and as a result, its acceptance within sports has grown as well. Demenius [3] looked at how managers, coaches, and assistant coaches felt about advanced data analytics, specifically in the Lithuanian Basketball League. They used a questionnaire to test this and found that overall, coaches and managers had a positive attitude toward basketball analytics and believed that it had a bright future. This research was conducted in 2017, six years after the movie, *Moneyball*, was released. It is clear that perception has changed regarding empirical analysis in sports, and therefore research on this topic has expanded.

Sugar and Swenson [15] looked at daily fantasy football and tried to generate lineups with positive expected returns. The data collected contained every game played since 2010 and resulted in about 40 statistics for every player in every game. They divided these features into three categories, recent history, current game, and career. They used forward selection to choose which features to include in the final model, splitting on position. They used Ridge Regression, Bayesian Ridge Regression, and Elastic Net and the Elastic Net provided the lowest root mean squared error. They then took those predictions and selected the DFS lineup using a binary linear program that will produce the maximum number of predicted points subject to the salary and position constraints. They also

suggested the idea of clustering players to improve future performance. Although this is a different sport, it addresses the same problem, creating an optimized lineup for daily fantasy sports. The methods that were successful here may not be successful in this paper as the statistics used will be different and therefore, the underlying relationship may also be different. Overall, the paper shows a good framework for approaching a problem like this.

Siglar and Compton [13] researched NBA players' salaries and how to relate that to the player's performance. Mainly, their focus was finding the player statistics that can be used to predict the salary cap room that will be given to each player on the team. They expanded on the work done by Lyons et al. [8], which found that points per game, rebounds, personal fouls, and field goal percentage are statistically significant in determining NBA players' compensation. Siglar and Compton looked to add two more variables that they thought were significant, 3-point shots made and player efficiency rating (PER). They used player data from the 2017-2018 NBA season. Using multiple regression, they found that points, rebounds, and assists were statistically significant at the .05 level and field goal percentage, 3-pointers made, blocks, and PER were not significant. Their research is applicable for the research question assessed here, as DFS lineups must fit the constraint of a salary. Features that are statistically significant in the case of the players real life salary would logically be significant in the salary assigned to them in a DFS lineup. Finding the statistics that are significant in predicting the DFS score could then be used to predict the players fantasy salary for the day, comparing it to the value assigned to them. This would show which players are less expensive than their expected value and which are more expensive than their expected value.

The research done by Dehesa *et al.* [2] looked to describe players' performances in NBA games using individual and team-based game variables. They used data from 535 games where the difference in score was less than or equal to eight points. They used cluster analysis and found five performance profiles. Three of these profiles were classified by their negative performance records. Classifying players into performance profiles would provide valuable knowledge about who that player is similar to. This approach could be expanded by creating a feature for the cluster that the player belongs to. This would then be used as input for the later models to predict the players fantasy points. In a similar fashion, some of the clusters could identify players that may have a negative impact on the final fantasy score.

3. Methodology

The data was collected using the nba_api [10] in Python. This API provides information on a large number of offensive and defensive statistics for individual games as well as season long averages for those same statistics. Each row in the dataset is one players statistics for an individual game. The final data set consists of 144,614 observations containing recent game features and game specific features. Since the idea is predicting the number of fantasy points a player scores in a game before the game starts, statistics such as points, rebounds, etc. cannot be used, since the game has not occurred yet. Some data wrangling must be performed to create some recent game features. To get around this, but still have a measure of how a player has been playing recently, rolling averages of these statistics over the previous 3, 5, and 7 games were created. Game specific features are also important to include since factors such as opposing team, number of days rest since last game, and whether the player is home or away will have a direct impact on the players performance. The final data set contains the 3, 5, and 7 game rolling averages for minutes, field goals made, field goals attempted, field goal percentage, 3 point field goals made, 3 point field goals attempted, 3 point field goal percentage, made free throws, free throw attempts, free throw percentage, offensive rebound, defensive rebounds, total rebounds, assists, steals, blocks, turnovers, personal fouls, points, and +/-. It also includes the opponent, days rest, and home.

Minutes refers to the number of minutes played by a player in a game. Field goals made refers to the number of 2 point and 3 point shots made by a player in a game. Field goals attempted refers to the number of 2 point and 3 point shots attempted by a player in a game. 3 point field goals made is the number of 3 point shots made by a player in a game. 3 point field goals attempted is the number of 3 point shots attempted by a player in a game in a game. A free throw is an unguarded shot from the free throw line given to a player following a foul or other infraction. A rebound occurs when a player retrieves a missed shot attempt. Offensive rebounds are the number of rebounds a player records while their team is on offense. Defense rebounds are the number of rebounds a player records while their team is on defense. Total rebounds is the number of offensive and defensive rebounds a player records in a game. Assists are passes that lead directly to a made shot. Steals refers to the number of times a player takes the ball away from the opposing team while playing defense. A block is recorded for a defensive player when they tip the ball while an opponent is attempting a shot, blocking their chance to score. A turnover is when a player on offense loses the ball to the defensive team. Personal fouls are the number of fouls a player commits in a game. Points refers to the number of points scored by a player in a game. +/- refers to how a team performs while that player is on the court. For example, if a player has a -5 as a +/- then that players team was outscored by the opposing team by 5 while that player was on the court. To create the opponent feature a dummy variable was created for every team, the variable for a specific team was set to 1 if that was the opposing team or a 0 if they were not. Days rest refers to the number of days since the players last game. Home is a binary variable containing a 1 if the game was played at home or a 0 if the game was played on the road [14].

The dependent variable, what is being classified, is fantasy points. DK calculates fantasy points using the values listed in Figure 1.

Scoring

Point	+1 Pt
Made 3pt Shot	+0.5 Pts
Rebound	+1.25 Pts
Assist	+1.5 Pts
Steal	+2 Pts
Block	+2 Pts
Turnover	-0.5 pts
Double-Double {Max 1 Per Player: Points, Rebounds, Assists, Blocks, Steals}	+1.5 Pts
Triple-Double {Max 1 Per Player: Points, Rebounds, Assists, Blocks, Steals}	+3 Pts

Figure 1: DK Scoring

This can be set up in the form of an equation.

$$Fantasy\ Points = Points + (0.5 * Made\ 3pt\ Shots) + (1.25 * Rebounds) + (1.5 * Assists) + (2 * Steals) + (2 * Blocks) - (0.5 * Turnovers)$$

Another 1.5 points are added onto this formula if a player records a double-double, which 10 or more in two of the categories listed under double-double in Figure 1. If a player records a triple-double, 10 or more in three of the categories listed, then an additional 3 is added to formula. The formula was used to calculate the fantasy points scored by a player each game. Instead of keeping it as a continuous variable, it was converted it into a binary classification problem. Fisher-Jenks natural breaks [12] was used to find the cut off value. Fisher-Jenks algorithm looks to classify n observations into k classes. It does this by looking to maximize variance between classes while minimizing variance within classes [12]. In this case the number of classes, or k, was set to two.

Three different models were used to classify fantasy points and compared the results. Those models are logistic regression [7], random forest [11] and an ensemble model which used logistic regression, a random forest, and a support vector machine (SVM) [16] with hard voting. Logistic regression is a statistical model that uses the log-odds for classification [7]. Random forest is a statistical model which utilizes multiple decision trees and takes a majority vote for classification [11]. SVMs create a boundary to separate the two groups based on the training data[16].

The same set of features was used on all three models. The data was split into a training set and a testing set with 80% of the data in the training set and 20% in the testing set. These were then saved to separate files so that each model would be using an identical training set and testing set. 10-fold cross-validation was used on the training set to determine parameter values for the models. Accuracy was used to evaluate the predictions on the test data. In terms of parameter selection, for the random forest model the number of trees was determined to be 1000 as it provided the highest average accuracy compared to the other values tried.

Overall, the solution can be summarized as:

1. Use API to obtain data.
2. Wrangle data to create recent game features.
3. Train the three models (logistic regression, random forest, ensemble model) on the training data.
4. Use the fit produced from each of these models to predict the test data.
5. Compare the models using accuracy.

4. Results & Discussion

The dependent variable, fantasy points, was converted to a binary variable using Fisher-Jenks natural breaks, as mentioned earlier. It was found that the value to split on was 21.0 fantasy points. So an observation is given a 0 if the amount of fantasy points that a player scored in that game is between -0.5 and 21.0. An observation is given a 1 if the amount of fantasy points that a player scored in that game is between 21.0 and 86.25. If a player scores exactly 21 points they will be given a 0 to be included in the lower scoring group. Figure 2 below shows the distribution of fantasy points as well as the cutoff points for the two levels. Looking at the entire data set, this leaves us with 75,062 observations with a 0 and 69,552 observations with a 1. This equates to 51.9% of observations containing a 0 and 48.1% containing a 1, so the classes are almost perfectly balanced

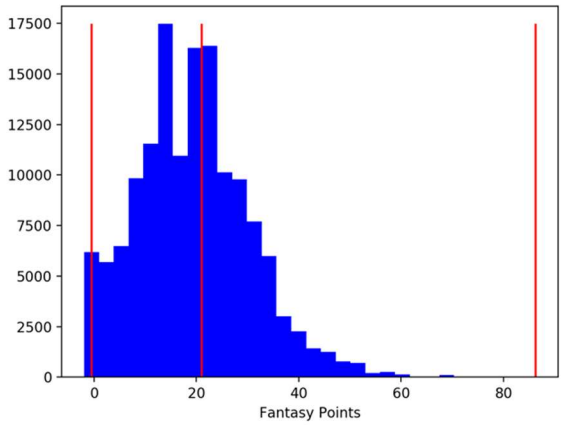


Figure 2: Fantasy Points Distribution

Since the same observations were used as the training data to train all three models as well as the same observations in the test data these models are directly comparable. Table 1 shows the results from each model. Sensitivity, also known as true positive rate, is the proportion of true positives (1) that are correctly identified as such. Specificity, also known as true negative rate, is the proportion of true negatives (0) that are correctly identified as such. Precision, also called positive predictive value, is the ratio of correct positive (1) predictions to total predicted positives (1).

Table 1: Results

Model	Accuracy	Sensitivity	Specificity	Precision
Logistic Regression	0.5385	0.364	0.699	0.525
Ensemble	0.5670	0.0.357	0.760	0.576
Random Forest	0.5839	0.446	0.710	0.585

The accuracy using logistic regression was 53.85%. Figure 3 below shows the confusion matrix for the logistic regression model. It shows that 0 was correctly classified more than 1 was. The model classified 19,342 as 0 and 9,581 as 1. The most incorrect classifications came from predicting a 0 when the true value was a

1, false negatives. The most correct classifications came from predicting a 0 when the true value was a 0, true negatives.

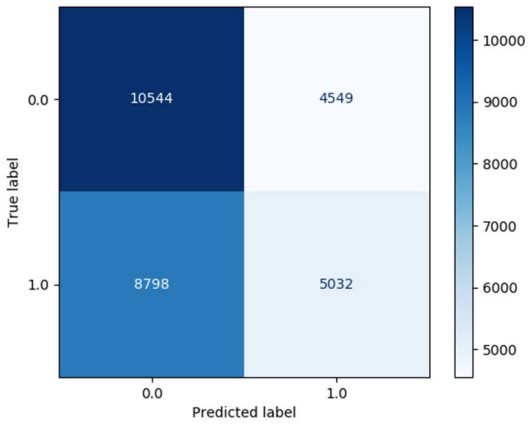


Figure 3: Logistic Regression Confusion Matrix

The accuracy using the ensemble model was 56.70%. Figure 4 shows the confusion matrix for this model. The model classified 20,365 as 0 and 8,558 as 1. The most incorrect classifications came from predicting 0 when 1 was the true value, false negatives. The most correct classifications came from predicting a 0 when the true value was 0, true negatives.

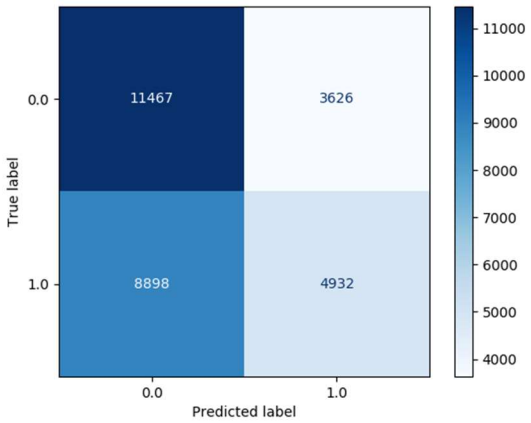


Figure 4: Ensemble Confusion Matrix

The accuracy using the random forest was 58.39%. Figure 5 shows the confusion matric for this model. The model classified 18,387 as 0 and 10,536 as 1. The most incorrect classifications came from predicting 0 when the true value was a 1, false negative. The most correct classifications came from predicting 0 when the true value was 0, true negatives.

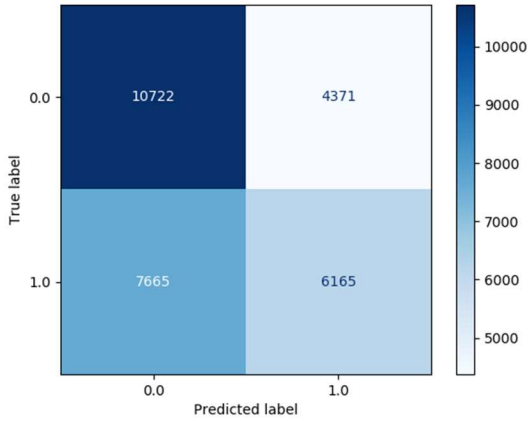


Figure 5: Random Forest Confusion Matrix

Overall, it can be seen that all of the models have more observations predicted as 0s than they do 1s and all the models predicted 0s more accurately than 1s. This can be seen directly in specificity as each's model's specificity is much higher than any other measure.

5. Conclusion & Future Work

DFS is difficult to predict and requires extremely high accuracy to reach the high levels of profit that some of the competitions offer. The accuracy of the best model here (57.82%) leaves something to be desired, but it can still be useful in setting your lineup. If you already play DFS and have knowledge of the sport then using these results as a secondary resource would help to improve your lineups. The models perform better at predicting when a player will score less than 21 points so keeping those players out of your lineup would be beneficial. If you combine the predictions from these models with the salary of each player, you can find the most benefit by finding players with very low salaries that are predicted to score over 21 points. This could still give you an edge over your opponent even if it only improves one position.

These models currently would not be able to create a full lineup, but they could be expanded in the future to do just that. The next step would be to look into finding more features related to the opposing team and players averages against that specific team. If this data or other models provide higher accuracy, you could expand from a binary to a multi-class problem, with the end goal being predicting the fantasy points as a continuous variable. After you can get that prediction, you could include salary and lineup restrictions and have the model return a full lineup with the highest predicted score.

6. References

[1] CHANGING THE GAME: The Rise of Sports Analytics: *https://www.forbes.com/sites/leighsteinberg/2015/08/18/changing-the-game-the-rise-of-sports-analytics/*. Accessed: 2019-11-19.

[2] Dehesa, R. et al. 2019. Key Game Indicators in Nba Players' Performance Profiles. *Kinesiology*. 51, 1 (Jun. 2019), 92–101.

[3] Demenius, J. and Kreivytė, R. 2017. The Benefits of Advanced Data Analytics in Basketball: Approach of Managers and Coaches of Lithuanian Basketball League Teams. *Baltic Journal of Sport & Health Sciences*. 1 (Jan. 2017), 8–13.

[4] DraftKings - Daily Fantasy Sports for Cash: *https://www.draftkings.com/help/rules/nba, https://www.draftkings.com/help/rules/nba*. Accessed: 2019-11-21.

[5] Industry Demographics • Fantasy Sports & Gaming Association: *https://thefsga.org/industry-demographics/*. Accessed: 2019-11-18.

[6] Lewis, M. 2003. *Moneyball : the art of winning an unfair game*. W. W. Norton.

[7] Logistic Regression Using SAS : Theory and Application: *http://eds.b.ebscohost.com.ezproxy.mercyhurst.edu/eds/ebookviewer/ebook/bmxlYmtfXzQ1MDk3NF9fQU41?sid=a4d6f0c1-dea1-498f-a06f-c18eca69da1e@pdc-v-sessmgr04&vid=3&format=EB&rid=2*. Accessed: 2020-04-14.

[8] Lyons, R. et al. 2015. Determinants of NBA Player Salaries. *Sport Journal*. (May 2015), 3–3.

[9] Miller, B. 2011. *Moneyball*. Columbia Pictures.

[10] Patel, S. *nba-api: An API Client package to access the APIs for NBA.com*.

[11] Python Data Science Cookbook: *http://eds.b.ebscohost.com.ezproxy.mercyhurst.edu/eds/ebookviewer/ebook/bmxlYmtfXzEwOTg5MDRfX0FO0?sid=5ffd8132-6b94-465d-85f8-8d4146cc90bb@sessionmgr101&vid=3&format=EB&rid=2*. Accessed: 2020-04-14.

[12] Rey, S.J. et al. 2017. An evaluation of sampling and full enumeration strategies for Fisher Jenks classification in big data settings. *Transactions in GIS*. 21, 4 (Aug. 2017), 796–810. DOI:https://doi.org/10.1111/tgis.12236.

[13] Sigler, K. and Compton, W. 2018. NBA Players' Pay and Performance: What Counts? *Sport Journal*. (Aug. 2018), 1–1.

[14] Stat Glossary: *https://stats.nba.com/help/glossary/*. Accessed: 2020-04-15.

[15] Sugar, G. and Swenson, T. Predicting Optimal Game Day Fantasy Football Teams. 6.

[16] Support Vector Machines : Data Analysis, Machine Learning, and Applications: *http://eds.b.ebscohost.com.ezproxy.mercyhurst.edu/eds/ebookviewer/ebook/bmxlYmtfXzU0MTg4OV9fQU41?sid=5ffd8132-6b94-465d-85f8-8d4146cc90bb@sessionmgr101&vid=17&format=EB&rid=1*. Accessed: 2020-04-14.

Automated Prediction of Voter's Party Affiliation using AI

Sabiha Mahmud Sumi
Department of Computing & Information Science
Mercyhurst University
501 E 38th St., Erie, PA, 16546
mahmud.sumi@gmail.com

ABSTRACT

The goal of this research is to develop the foundation of a cross-platform app, Litics360, that helps political election campaigns utilize data-driven methods, and high-performance prediction models, to align candidates with voters who share similar socio-political ambitions. To attain this goal, the first step is to understand a voter's currently aligned political party affiliation, based primarily on historical records of their turnout at previous elections, and basic demographic information. This research paper aims to find a solution to this first step, by comparing varied performance measures to find a reliable prediction model from learning algorithms, including decision tree, random forest and gradient boosting machine XGBoost binary classifiers. Significant correlations between independent variables and the target prediction class, i.e., voter's registered party affiliation, contribute towards the development of an automated predictive ML model. The Ohio Secretary of State public voter database was used to collect voter demographics and election turnout data, then prepared using preprocessing methods, and finally used to identify the best performing ML model. Hyperparameter grid search with XGBoost proved to be the superior binary logistic classifier, reproducing a nearly perfect skillful model. Tracking the alignment between voters and PEC candidates the proposed future of Litics360; i.e., to develop an application that promotes a healthy and transparent platform for voters to communicate their socio-political grievances to PECs, enabling efficient appropriation of a PEC's funds and resources to engineer successful marketing campaigns.

Keywords

Political election campaign; Voter support prediction; Voter party affiliation classification; Voter recommendation application; Election prediction; Political marketing; Decision tree classification; Random forest classification; Hyperparameter grid search with XGBoost; Gradient boosting classification; Binary logistic classification; Machine learning; Optimized prediction algorithm; AI;

7. INTRODUCTION

The primary goal of any PEC is to increase the probability of victory. To reach this goal, all aspects of a PEC must be evaluated based on the efficiency and cost of a campaign or activity, i.e., will it motivate desired voters enough to cast their votes. This cost-to-benefit analysis can be an efficient process, especially when driven by high-performance ML models that predict specifically aligned sets of voters, with whom a political candidate may share their promises and messages with, and engage them further in joining their socio-political missions. Therefore, such predictions are crucial for promoting better communication between voters and their politically aligned candidates.

With each passing year, PECs in the United States are rapidly seen adopting various data-driven targeted-voter recommendation applications that are tailored to reduce the cost and increase the benefit in maximizing the likelihood of their victory [5]. The spread and access of big data has significantly improved the quality of empirical analysis in finding, aggregating and identifying patterns and similarities across several domains. For example, recent successes of data-driven PECs have heavily relied on using knowledge from several domains (political, economic, medical-sciences, marketing etc.) to find novel patterns that suggest more efficient communication strategies with [20]. The underlying concept here is what the no-free-lunch (NFL) theorem suggests, that seeking a universal algorithm that solves all optimization problems is impossible, indicating that an optimization algorithm will perform better for some optimization problems better than others [32]. To that end, NFL may also be considered to suggest that the success of any data-driven target-voter recommendation app, requires unique integrations and applications of searching and optimizing that are developed from novel treatments of learned strategies across multiple domains.

The goal of this project is to develop a prediction model that accurately identifies any given voter's party affiliation based on their voting history, basic demographics, and their associated congressional district representative's party affiliation and CPVI score. The larger conceptual framework of this model is geared towards becoming a website-and-mobile-friendly platform, called LITICS360, that ultimately promotes the collection, archival, and retrieval of PEC and voter data through app-use by the PEC staff as well as voters. LITICS360 as a cross-platform and responsive app, aims to facilitate transparent communication between PEC candidates and their voters, to promote fair competition between all PECs alike. The principle of LITICS360 is motivated by the First Amendment of the U.S. Constitution, "where the democratic right of every U.S. citizen guarantees the ability to exercise freedom of speech … and the right of the people peaceably to assemble and consult for their common good, and to apply the Government for redress of grievances" [31]. The ideals behind successful democratic governments are historically marked by transparent communication of voter grievances on socio-political issues and a PEC candidate's promises to address those issues through promises of political change.

The underlying requirements for achieving the idealistic and conceptual framework for LITICS360 can be divided into four general stages for prototyping:

i. Primary predictive model that will accurately identify a voter's party affiliation based on turnout history from previous elections, basic demographics, and their

congressional district representative's party affiliation and CPVI score;

ii. Secondary predictive model that will assign a CPVI score indicating a voter's likelihood to support their affiliated party, based on answers to survey questions and polls from marketing campaigns conducted by PECs;

iii. Tertiary prediction model will evaluate their calculated CPVI score and predicted party affiliation, to further indicate a voter's likelihood to turnout at an upcoming election, based on their positive or negative interactions with PEC activities, i.e., page views and visits for a candidate's landing site, PEC event/rally attendance, donations, and direct communication with PEC and more;

iv. Finally, these models will be deployed into a cross-platform app, that is developed with user experiences (UX) engineered from voter data and feature inputs by PECs, and delivered on an interactive user interface (UI) that visualizes the predictive analyses and competitive intelligence in real-time, for PEC candidates, their staff, and their voters.

This paper will be focusing on building the first stage of the larger conceptual framework of LITICS360, i.e., Pv1.0. While several dominating proprietary apps, as well as research papers, boast novel prediction models and share similar conceptual foundations, there will always be room for accommodating classification and clustering techniques that help uncover newer and more efficient patterns and similarities. Litics360 Pv1.0 aims to do just that, i.e., to contribute a model that accurately predicts a voter's party affiliation.

8. paper Organization

This paper aims to develop the groundwork for Pv1.0, which includes the following stages:

i. **Research:** Historical background, evaluation of related works and apps;

ii. **Data:** Collection, variable measures, feature variable selection and engineering, and wrangling methodologies;

iii. **Model:** Architecture, performance measures, optimization tuning of hyperparameters;

iv. **Evaluation:** Model and performance results;

v. **Conclusion:** Comparison of models and concluding remarks

vi. **Future work:** Signification of this research and future work for building Litics360 app

9. background & Related works

The culture of US-based PECs has dramatically evolved over the past decade. Historically (i.e., 1800s, 1900s and early 2000s), PECs and affiliated parties were known to use rudimentary standards for predicting voter turnout and support tendencies, in comparison to data-driven and media-powered communication standards used over the past decade. Turnout and party support predictions were previously based on the historical performance of precincts over the past four general elections, primarily measured by percentage of votes for any given party. Consequently, PEC-to-voter communication strategies would consist of re-connecting with previous donors and volunteer captains. Campaign strategies are more reliant on numbers-driven campaigns, implying poll numbers and policies in response to manually-recorded surveys [11].

In the recent decade, previously used numbers-driven campaigns have evolved into data-driven campaigns. Scientific research of efficient data processing and improved computational power has introduced the notion of big data. Dynamic databases, improved analytic methods and development of prediction models using ML, are increasingly becoming specialized to compete with current PEC market standards. The 2012 and 2016 presidential elections saw the proliferation of data-driven PECs (for candidates Barak Obama vs. Mitt Romney and Donald Trump vs. Hillary Clinton). This big data revolution has not radically transformed PECs as much as television did in the 1960s, but in a close political contest, data-driven strategies can have enough impact to make the difference between winning and losing [22].

Powering existing PEC strategies with smart data-driven technologies hasn't been a popular choice, when state or county level elections are concerned. Inspiring grass-root and local PECs to adopt these data-driven prediction models require the app's front-end interfaces to be a brand-centric user-friendly experience that is appealing to the user. The bottom line for the mass-adoption of any novel PEC prediction algorithm is to keep in mind that the first customer of these apps will be the campaigners themselves. Therefore, the app must have an integrated approach to promote user-friendly experiences and interfaces to market the power of PEC prediction models using ML.

9.1 Evaluation of reviewed published works and apps

9.1.1 Big data-driven classification approaches

The primary concern with any given data-driven approach for PECs is the accuracy of prediction scores in forecasting the behaviors, preferences, and responses of voters. The secondary concern is a measure of the practicality of the app and its user-friendliness. A simplistic guide to successful predictive scoring models is to focus on creatively and critically thought-out variables that are sensibly linked to interesting predictions with the empirical validity [22].

There are various data-driven approaches to classifying voters that several research papers have identified. Table 1 categorizes several relevant works into five different approaches: Voter demographics; voter behaviors and preferences through means of surveys and polling; voter responsiveness through voting history, events, donations, marketing and advertising; and sentiment analysis through social media.

#	Classification approaches	Reviewed published works, citations
1	Classifying voter demographics	[2] [24] [23] [17]
2	Classifying voter behaviors and preferences (surveys, polling)	[8] [2] [5] [1]
3	Classifying voter responsiveness (voting history, events, donations, marketing/advertising)	[19] [15] [2] [3]
4	Sentiment analysis (social media)	[13] [5] [12] [17] [18] [4]

Table 1: Reviewed published works categorized by approaches for classification

The first classification approach discusses voter demographic information available in registered voter files such as name, gender, age, geolocation, county, congressional district, registered party affiliation, among other basic demographic information. Several research papers, as shown in Table 1, have highlighted that these models simply predict or classify based on focal traits of interest, rather than why they voted/donated or showed support for a candidate. As such, figuring out causation

is not the biggest concern, rather prediction accuracy is the main goal [2] [24] [23] [17].

The second type is based on the behaviors, attitudes, and preferences of voters to reveal predictive scores for behavior or support scores based on surveys and polls. Several research papers, as shown in Table 1, have highlighted that these models too do not make causal claims about why the people turned up to vote or why they supported a particular candidate. Again, the intention is similar to the first category, i.e., to avoid overfitting the data [8] [2] [5] [1].

The third type is based on voter responsiveness to marketing or advertising campaigns, participation in events, rallies, demonstrations, giving donations, all in coordination with their turnout history. Research works, as highlighted in Table 1, have discussed these responsiveness scores, most of which are heterogeneous reactions to PECs in a randomized voter base as well as media-based voter base. The results derived from the effect of such marketing strategies, are used as important variables that influence further communication strategies of PECs. Strong/weak positive/negative responsiveness is not causal to how the campaign was moderated or streamlined. Rather, it is about the observed differences and search for correlations across many subjects and variables used in the forming of messages, that generate interesting results [19] [15] [2] [3].

The fourth type is based on text-based classification and analysis of publicly expressed sentiments on social media. Research works, as highlighted in Table 1, have discussed such sentiments to provide probable causal links of voter's candidate support score based on standings (i.e., for/against) on political issues. Opinion mining and sentiment analysis have rapidly become a popular data-driven approach for machine learning. However, more often than not, baseless causal links are grounded on the theoretic rationale of the model's architecture [13] [5] [12] [17] [18] [4].

9.1.2 ML algorithms and prediction models

Currently, the vast majority of the predictive scores used by PECs are created by a PEC data analyst (or a team of them) using simple regression techniques: Ordinary least squares for continuous outcomes; logistic regression for binary outcomes; and, rarely, to bid for truncated data like dollars donated or hours volunteered [8]. A wide variety of skills are needed for developing such models customizing them to specific political environments.

PEC data analysts have been searching for more systematic methods for selecting a preferred regression. The commercial marketing industry often uses k-means clustering or k-nearest neighbor classifier to divide consumers into categorical types such as blue collar, grilling, SUV owners. However, such methods of clustering data based on voters or families are not as useful anymore for campaign data analysts, as strategic decisions in campaign planning are reliant on cost-to-benefit analysis and person-specific probabilities for particular outcomes. Thus, knowing that a set of citizens are similar in many dimensions does not assist with PEC-to-voter communication, if those dimensions are not highly correlated with voting behaviors, ideology, and propensity to donate [26] [5] [7] [8].

Supervised machine learning includes methods such as classification and regression trees. In a regression tree approach, "the algorithm grows a forest by drawing a series of samples from existing data; it divides the sample based on where the parameters best discriminate on the outcome of interest; it then looks at how regressions based on those divisions would predict the rest of the sample and iterates to a preferred fit" [6]. The payoff for this approach is that it generates estimates of what parameters are most important: that is, what parameters add the most predictive power when the group of other parameters is unchanged [26].

Other methods such as support vector machine (used to find maximal geometric margins that separate positive from negative predictions) and naïve bayes (predicting the likelihood of seeing feature vector when conditionally independent) are often used in supervised ensemble learning models to find correlations between census demographics and voter behavior features for support prediction [8] [26] [5] [13].

ML algorithms/models	Reviewed published works, citations
Logistic regression (LR)	[8]
Natural language processing (NLP)	[13] [5] [12] [17]
K-nearest neighbor classifiers (kNN)	[26] [5] [7]
Random forest classifiers	[26]
K-means clustering (K-means)	[8]
Support vector machine (SVM)	[8] [26] [5] [13]
Naïve Bayes	[26][13]

Table 2: Reviewed published works categorized by learning algorithms and models

In Table 2, it can be seen that natural language processing (NLP) algorithms have become increasingly popular. A great many of the published works investigate the potential benefits that NLP brings to finding etymological trends and harnessing the power of such trends to motivate the public towards specific PEC strategies

Compared to previously published work on PEC support and turnout probability prediction, the novel contribution of this paper is to combine voter demographic, behavior and responsiveness for predicting scores for high and low probabilities of voter turnout. In addition, I will investigate the use of ensemble learning methods will allow for more finetuning and organization of the predictive path rather than using a random search predictive model.

9.1.3 Reviewed existing proprietary apps

App	Plans/platforms/sub-apps	Citation
Ecanvasser	Essentials, Walk app, Go app, Leader	[10]
i-360	Walk, Call, Portal, Field Portal, Action, Text, Vote	[14]
L2 Political	Voter mapping, Constituent mapping, Voter outreach lists, Email/texting deployment, Digital advertising, Consumer Mapping, Automapping, V-count, Printable reports	[16]
Nation Builder	Software, Non-profit network, Enterprise organization, Advocacy, Politicians	[21]
The Optimizer	Native, Mobile, Display	[29]

Table 3: Reviewed published works categorized by existing proprietary applications

Several existing proprietary apps, as briefly highlighted in Table 3, are available in the PEC app market that feature different capabilities including:

i. Outreach capabilities: Canvassing, survey building, casework management, landing-website templates, targeted emails, and fundraising;
ii. Analytic/statistic capabilities: User-activity and app-use, statistics, a breakdown of community information,

databases with map views and filters, email statistics (opens, clickthroughs, bounces, unsubscribes and spam reports), social media data capture, demographic data analytics, twitter-inferred political score, and data-driven similarity scores between comparing individuals/donors/volunteers/ event-attendees using ML models;

iii. Organization capabilities: Integrity of local or national campaign data with sub-campaigns for staff with a unified system for teamwork, database synced to supporter profiles and interactions, custom reporting, dashboards, goal setting/tracking/measuring.

Ecanvasser, a political canvassing campaign app, is primarily designed to help PEC organize their efforts by syncing electoral registers between its dashboard and canvassing mobile app, in addition to the management of issues and advocacy groups for community engagement through grassroots mobilization tools. Ecanvasser is used mainly by PECs that are in election mode or for managing constituency work. [10].

i-360 is a solution for political campaigning, non-profits and organization with grassroots technology that integrates management system and database for predictive models, digital/TV communication, and real-time analytics [14].

L2 Political is a PEC app that boasts a national voter file and selection platform at its core. This data-driven app provides access to their comprehensive voter-data to varied customers including: Local/state/federal campaigns, general consultants or direct response or media pollsters, and organizations such as (PACs/Super PACs/Associations/Unions). Their database is powered by five types of information records including: Voters with predictive attributes, non-registered voters, consumers, and constituents. The database enables voter mapping and filtration through varied demographic/psychographic/behavioral attributes that boast 600 behavioral, 400 demographic and 91 predictive data fields. Outreach capabilities include: Mailing lists, canvassing/walk lists, phone/text lists, email lists, and digital communication [16].

Nation Builder (NP) is an app that is developed to facilitate organizations, movements, and campaigns alike. NP promotes several plans (primary, non-profit network, enterprise organization, advocacy organization and political campaign) with capabilities for outreaching, measuring analytics and campaign organization [21].

The Optimizer is another performance-based automated optimization app for native advertising, push/pop/redirect traffic campaign optimization process, and banner campaigns [29].

10. Data & methodology

10.1 Pv1.0 Problem Scope Definition

Pv1.0 aims to contribute a voter party affiliation identification model using learning algorithms. Independent variables include, the voter's id number, residential markers, basic demographics, election turnout history between years 2000 and 2020. It is important to also note that a voter's support for their congressional district incumbent is critical to understand whether a voter's stance on political issues (for or against) is aligned with that of their district representative. This is why it is import to also include voters' congressional district representative's affiliated party and CPVI score as another independent variable. The aim is to accurately predict a voter's party affiliation, as it is the target variable.

10.2 Data collection, feature selection and engineering

10.2.1 Data Selection Categories and Rationale

i. Voter identity and election turnout history:

- To develop a highly accurate party affiliation identification prediction model, a large database of voter information is required. The Ohio Secretary of State's statewide voter demographic database hosts publicly available datasets containing demographic and historical election turnout information of over seven million voters in the state [28].
- The database is a public record collection of registered voters in the state of Ohio, as submitted by each county Board of Elections. These records are submitted and maintained in accordance with the Ohio Revised Code for access to and use of voter registration lists which are open to public inspection and use for non-commercial purposes [27].
- Election voting history of the voters', for primary and general elections from year 2000 to 2020 as provided by the counties [28]. Voter turnout history is neither complete for each election nor for their party of choice.

ii. Ohio's congressional district representative's party affiliation and CPVI score:

- Using a simple website scrapping method, the congressional district representative's party affiliation and CPVI score were mined from the official US House of Representatives directory [9]. The scrapped data was then merged with each voter data using their associated congressional district number.

10.2.2 Feature selection and engineering

Preprocessing of all datasets has been done using Python and related libraries using Jupyter Notebooks. Data wrangling and cleaning applied to the Ohio voters' dataset, includes the steps described below:

- Voters missing data for voter ID, congressional district and county number were dropped as such information is vital to creating a model for the model.
- Demographic features with over 90% missing values were also dropped as these were not important features.
- Voter's date of birth and registration date were converted to datetime then converted to a numeric age value.
- Features that were extracted from this dataset included each voter's: Voter ID, county number, DOB, party affiliation registration date, voter status, party affiliation, voter residential zip code and congressional district number.
- Features containing voter's turnout between years 2000 to 2020 for general and primary elections were also extracted from the dataset. The values for these features represent their voting history and election turnout over the years.
- All values related to political parties, i.e., election turnout features, congressional party affiliation and voter party affiliation, were converted to numeric values using a conversion key demonstrated below in table 4.

Converted numeric value	Party affiliation marker	Party affiliation definition

0	D	Democratic party
1	R	Republican party
2	C	Constitution party
3	E	Reform party
4	G	Green party
5	L	Libertarian party
6	N	Natural Law party
7	S	Socialist party
8	X	Voted without declaring party affiliation
9	NA	No voting record

Table 4: Conversion key for parties to numeric values

10.2.3 Target Variable for Prediction

Pv1.0 model is a party affiliation identification prediction model, so the target class variable here is the registered party affiliation of the voter. Voters missing the records of their registered party affiliation were dropped, as the aim here is to develop and train a model that predicts the party affiliation. As the US presidential political system is more of a bipartisanship shared by the republican and democratic parties, the dataset was further stripped of any voters whose registered party affiliation was not republican or democratic.

10.3 Data preprocessing and splitting

After cleaning and wrangling of the data, the data was first divided for modeling, training and validation. The dataset included the information of a total of 7,772,371 registered voters from the state of Ohio. Among these, the number of voters who declared their registered party affiliation is a total of 3,207,039. The total dataset was split into two sets, a model set for building, training, and validation of the ML models (containing ninety percent of total or 6,994,611 voters), and a blind set for testing of the final model (containing ten percent of total data or 777,179 voters).

11. pv1.0 model

This section provides brief backgrounds on the three learning algorithms (i.e., decision tree classifier, random forest classifier and gradient boosting XGBoost classifier with hyperparameter grid search) utilized in this paper to accurately predict the voter's registered party affiliation.

11.1 Decision Tree Classifier

Decision tree classifier is one of the most powerful and popular algorithm, falling under the supervised learning algorithm umbrella [25]. The algorithm works great with categorical target variables as they are classified and sorted down from the root to the leaf node, providing a classification at each level.

In a binary tree, like the one developed for this case, classifiers are constructed by repeatedly splitting subsets of the learning sample into two descendent subsets, beginning with the learning sample itself. To split the learning sample into smaller subsets, the splits have to be selected in such a way that the descendent subsets are always purer than the parents [25]. The impurity function is based on the gini index criterion, which selects a test sample that maximizes the purity of the split. The information gain function is based on the entropy criterion, which selects a test that maximizes information gain [25]. These functions take the form of the equations in figure 1:

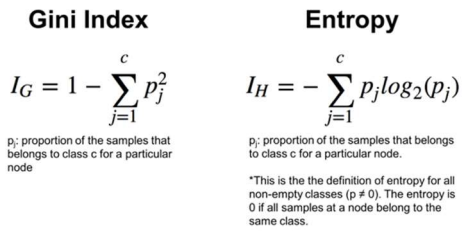


Figure 1: Criterions – Gini index and entropy

Figure 2 below illustrates the binary tree developed using the decision tree classifier. This particular tree uses the gini index criterion to show the purity of the classification from the root node, which is the feature of voter's turnout for primary election on March 15, 2016. In 864,050 voter test samples' predicted classes at the max_depth of 1.

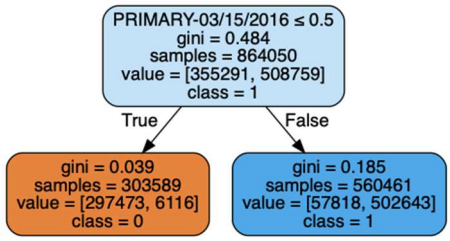


Figure 2: Decision tree result using gini criterion and max_depth of 1 for test samples

11.2 Random Forest Classifier

Random forest classifier is an ensemble algorithm that combines more than one algorithm, in this case the decision tree algorithm, to classify objects. Furthermore, ensemble learning creates a set of decision trees (each of which is trained on a random subset of the training data), used to create aggregated predictions providing a single prediction from a series of predictions. The advantages include: Highly flexible and very accurate, naturally assigns feature importance scores, so can handle redundant feature columns, has the ability to address class imbalance by using the balanced class weight flag, scales to large datasets, generally robust to overfitting, data does not need to be scaled, and can learn non-linear hypothesis functions. The disadvantages include: Results may be difficult to interpret, the importance each feature has may not be robust to the variations in the training dataset [6].

11.3 Gradient Boosting Classifier using XGBoost

Gradient boosting classification is a sequential technique, which works on the principle of an ensemble. It combines a set of weak learners and delivers improved prediction accuracy. A weak learner is one, that is slightly better than random guessing. At any instant t, the model's outcomes are weighed based on the outcomes of previous instant t-1. The outcomes predicted correctly are given a lower weight and the ones miss-classified are weighted higher [30].

The effect of this is that the model can quickly fit, then overfit the training dataset. A technique to slow down the learning in gradient boosting is to apply a weighting factor for the corrections by new trees when added to the model, i.e., with XGBoost. When creating the gradient boosting model, XGBoost is a great tool for tuning the learning rate hyperparameters to control the weighting of the new trees added to the model. Grid

search capability using scikit-learn can be used to evaluate the effect on the logarithmic loss of training a gradient boosting model with different learning rate values [30].

11.4 Performance Measures

The three learning algorithms, i.e., decision tree, random forest, and gradient boosting classifiers, help in developing the Pv1.0 model.

The decision tree classification model varies gini and entropy criterion as well as max_depth ranging from 1-9. The model's accuracy is defined as the fraction of correct predictions out of total number of data points. Finding the optimal value for max_depth is the tuning method used to find the best accuracy score and receiver operating characteristic (ROC) area under the curve (AUC) accuracy score.

The random forest model uses n_estimators to set a number of trees and uses ROC curves and precision and recall scores show the probabilistic forecast for this binary classification model. These performance measures use values from both columns of the confusion matrix to evaluate the fraction of true positives among positive predictions.

Gradient boosting with XGBoost model varies hyperparameters with a range of learning rates from 0.01 to 1.0 and n estimators of 10 and 100 to find the best tuned learning algorithm using the grid search method. The model's accuracy is defined using the ROC AUC score plotted to show the rate of true positives, which is the fraction of the elements of 1 that are classified as 1 correctly, as a function of the false positive rate, which is the fraction of the elements of 0 that are classified as 0 incorrectly. The sensitivity is given by the rate of true positives and anti-specificity by the rate of false positives. The anti-specificity, or false positives, correlates with the x-axis, and the sensitivity, or true positives, correlates to the y-axis, which forms the ROC AUC figure displayed. A subset of the data is also chosen through the confusion matrix for measuring the quality of the classification system.

12. pv1.0 model evaluation

The model developed for Pv1.0 uses the decision tree classifier at first in order for an easier demonstration of the model. Random forest model helps to create better evaluation on principle as it combines a number of weak estimators to form a strong estimator. Finally, the gradient boosting model with XGBoost is the solution model using grid search to tune hyperparameters for higher performance with higher learning rates and a larger number of trees. In this section, the performance results of these three models will be discussed.

12.1 Performance Results

12.1.1 Decision Tree Classification Model

The decision tree model had a range of max_depth inputs, in combination with gini and entropy criterion. The performance accuracy as well as the ROC AUC accuracy were calculated for each hyperparameter variation, to find the best combination that resulted in high accuracy and ROC AUC accuracy scores.

Max_depth	Criterion	Accuracy	ROC AUC accuracy
None	gini	0.99959	0.99958
None	entropy	0.99966	0.99965
1	gini	0.92643	0.91267
1	entropy	0.92643	0.91267
2	gini	0.98659	0.98959
2	entropy	0.98659	0.98959
3	gini	0.99316	0.99680
3	entropy	0.99316	0.99680
4	gini	0.99760	0.99856
4	entropy	0.99760	0.99856
5	gini	0.99918	0.99914
5	entropy	0.99915	0.99914
6	gini	0.99930	0.99928
6	entropy	0.99918	0.99984
7	gini	0.99930	0.99932
7	entropy	0.99925	0.99995
8	gini	0.99938	0.99945
8	entropy	0.99926	0.99997
9	gini	0.99939	0.99987
9	entropy	0.99926	0.99997

Table 5: Decision tree model performance results for model validation dataset

Table 5 below contains the performance measure of the decision tree with varied max_depths and criteria gini and entropy. Based on the performance results laid out in table 5, it can be seen that with higher max_depth the accuracy increases.

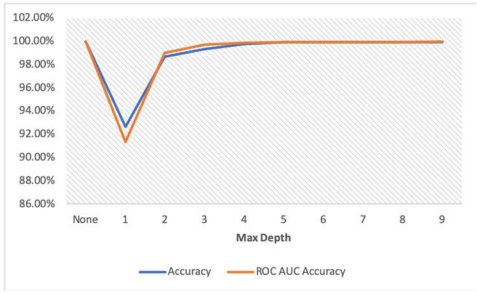


Figure 3: Decision tree model performance results for test samples of model validation dataset using gini criterion

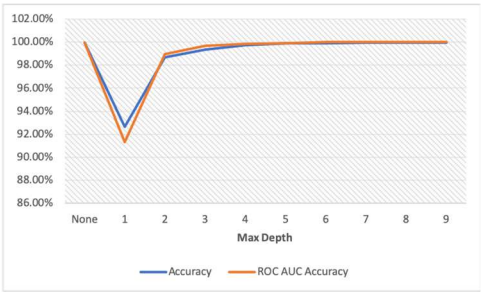


Figure 4: Decision tree model performance results for test samples of model validation dataset using entropy criterion

Figure 3 shows the performance results for gini criterion for the range of max_depths, and figure 4 shows the same for entropy criterion. The difference between gini and entropy isn't far apart, as it can be seen in the figures 3 and 4. The performance accuracy as well as the ROC AUC accuracy scores are at their highest when max_depth ranges from 5 and above.

12.1.2 Random Forest Classification Model

The random forest model was set to run with n_estimators set to 100 trees. Table 6 below contains the performance measures of the random forest learning algorithm, including recall, precision, and ROC scores, which show the probabilistic forecast for this binary classification model.

Score type	Baseline	Train samples	Test samples
Recall	1.0	0.9936	0.9941
Precision	0.5896	0.9978	0.9979
ROC	0.5	0.9999	0.9999

Table 6: Random forest model performance results for model validation dataset

The results listed in table 6 show the recall, or sensitivity, score where the ratio of correctly predicted positive observations to the total predicted positive observations is at its highest level for both training and testing samples. These results show that of all the registered voters' party affiliation was accurately predicted the in the random forest model. Table 6 also shows the precision score where the ratio of correctly predicted positive observations to the total predicted positive observations is also at its highest level for both training and testing samples. This high and near-perfect precision score communicates the low false positive rate.

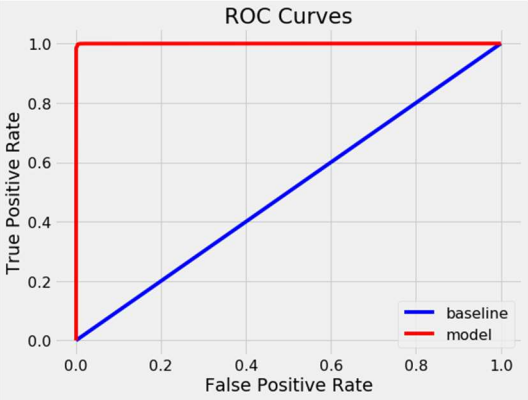


Figure 5: Random forest model ROC performance results for model validation dataset

Figure 5 plots the false positive rate on the x-axis versus the true positive rate on the y-axis for a number of different candidates', where the threshold values fall between 0.0 and 1.0. In figure 5, the false alarm rate is compared with the hit rate, which is demonstrated by the difference in the rates of true positives and false positives. The true positive, or the sensitivity, is calculated as the number of true positives divided by the sum of the number of true positives and the number of false negatives. The main takeaway from figure 5 is how well the model that is predicting the positive target class of voter's party affiliation when the actual outcome is also positive. The false alarm, or the inverted specificity, rate summarizes how often a positive class is predicted when the actual outcome is negative. This inverted specificity is the total number of true negatives divided by the sum of the number of true negatives and false positives.

Represented at a point (0,1), figure 5 shows a line travelling from the bottom of the left of the plot to the top left and then across the top to the right. This representation of line shows a nearly perfect skillful model, where the probability of randomly chosen real positive occurrences versus negative occurrences, is at its highest.

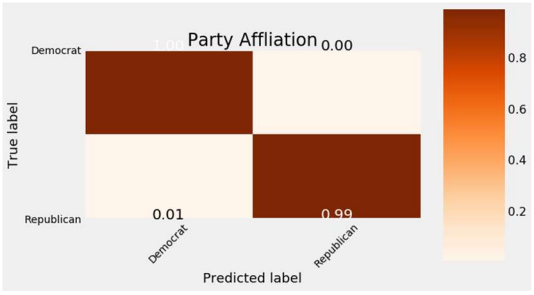


Figure 6: Normalized confusion matrix for random forest model with model validation dataset

The ROC curve in figure 5 plots a near-perfect skillful model, which is also demonstrated in figure 6's confusion matrix. Figure 6 displays the confusion matrix table describing the performance of the random forest classification model on the test data samples for which the true values are known. True positives and true negatives are the observations that are correctly predicted, shown in reddish-brown color representing a close to 1.0 accuracy.

12.1.3 Gradient Boosting Classification Model with XGBoost Using Grid Search for Tuning

The gradient boosted classification model with XGBoost was set to run with n_estimators set to 10 trees and 100 trees. This model also had a range of learning rates including 0.001, 0.001, 0.025, 0.05, 0.075, 0.1. These hyperparameters were fitted using grid search to find the best XGBoost classifier.

Figure 7 shows negative logarithmic loss for hyperparameter grid search with n_estimators from 10 to 100 trees and varied learning rates, where the loss function is used to quantify the price paid for inaccuracy of predictions within this classification. Figure 7 shows that the best mean negative log loss score was -0.004291 with a standard deviation of 0.000002 for the hyperparameters of 0.1 learning rate and 100. Negative log loss performance results were calculated by doing a randomized grid search for the XGBoost classifier with varying hyperparameters of n_estimators and learning rates states above.

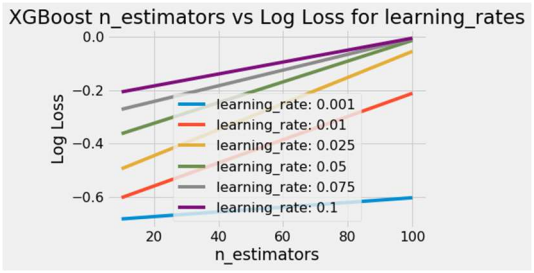


Figure 7: Negative log loss results using hyperparameter grid search for XGBoost model

Table 7 below contains the performance measures of the XGBoost model, including recall, precision, and ROC scores, which show the probabilistic forecast for this binary classification model. These scores were calculated after applying the best estimated hyperparameters for XGB classifier, using the grid search mentioned above.

Score Type	Baseline	Train Samples	Test Samples
Recall	1.0	0.9998	0.9998
Precision	0.5896	0.999	0.9989
ROC	0.5	0.9999	0.9999

Table 7: XGBoost performance results for model validation dataset

The results listed in table 7 are very similar to the random forest model as it shows the recall, or sensitivity, score, where the ratio of correctly predicted positive observations to the total predicted positive observations, is at its highest with both train and test samples. This high recall score indicates that of all the registered voters' party affiliation were accurately predicted the in the XGBoost model. The precision performance measure shows the ratio of correctly predicted positive observations to the total predicted positive observations. As stated in table 7, the high and near-perfect precision communicates the low false positive rate.

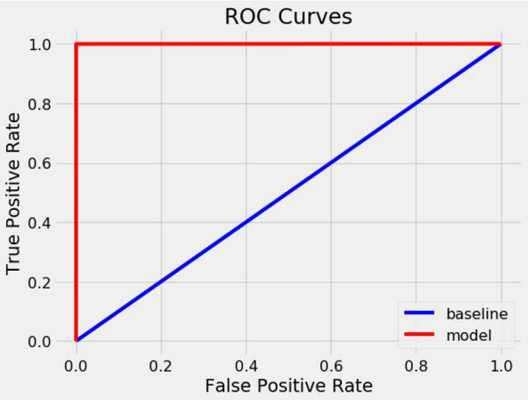


Figure 8 XGBoost ROC Results for model validation dataset

Figure 8 shows the ROC curves for the XGBoost model which is very similar to figure 5 as the model results are near perfect in both the random forest and the XGBoost models. Again, represented at a point (0,1), figure 8 shows a line travelling from the bottom of the left of the plot to the top left and then across the top to the right. This representation of line similarly shows another perfect skillful model, where the probability of randomly chosen real positive occurrences versus negative occurrences, is at its highest.

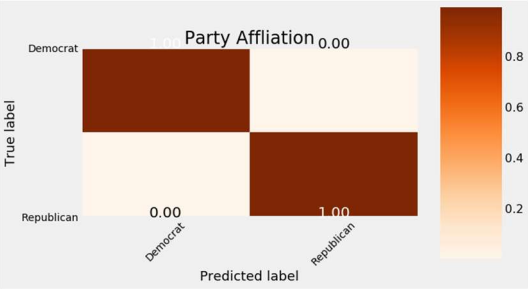


Figure 9: Normalized confusion matrix for XGBoost model for model validation dataset

Figure 9 displays the confusion matrix table describing the performance of the XGBoost classification model, which is similar to the confusion matrix of the random forest model shown in figure 6. As the ROC curve in figure 7 plots a perfect skillful model where the true positives and true negatives are predicted accurately, figure 8 reconfirms that the true positives and true negatives are correctly predicted, shown in reddish-brown color representing a close to 1.0 accuracy.

12.2 Feature Importance

In order to increase the quality of predictive power of the learning models against the binary target of party affiliation, finding the variables that are strongly correlated with the target class is important. Table 8 shows the top three features, or the most predictive variables, that influence the highest accuracy performances of the binary classification predictions in the decision tree, random forest, and XGBoost models. It can be seen here that the feature that has the strongest correlation to the target class, is the voter turnout at the primary election on March 15, 2016, with an importance score of over 70% for both decision tree and XGBoost models and over 50% for the random forest model.

Feature column number	Feature name	Feature importance score
Decision tree model		
53	PRIMARY-03/15/2016	0.724
60	PRIMARY-05/08/2018	0.246
63	PRIMARY-05/07/2019	0.020
Random forest model		
53	PRIMARY-03/15/2016	0.509
60	PRIMARY-05/08/2018	0.237
26	PRIMARY-03/04/2008	0.048
Gradient boosting with XGBoost using grid search model		
53	PRIMARY-03/15/2016	0.706
60	PRIMARY-05/08/2018	0.166
63	PRIMARY-05/08/2018	0.189

Table 8: Top three features influencing prediction models, in their order of importance

12.3 Testing XGBoost model with blind test dataset

With performance results of over 99% accuracy in all three models for the model validation dataset, the next step is to utilize the apply the best ML model, while increasing the quality of its predictive power as well. This can be done by removing the more influential feature within the dataset i.e., the feature with the maximum importance score for all three models, from the test dataset. After which, the best estimated hyperparameters (i.e., learning rate of 0.25 and 100 n_estimators) that were found using the randomized grid search, were applied to calculate the XGBoost model's performance scores. Table 9 below contains the performance measures of the XGBoost model, including recall, precision, and ROC scores, for the blind test dataset.

Score type	Baseline	Train samples	Test samples
Recall	1.0	0.9274	0.9257
Precision	0.5895	0.8793	0.8769
ROC	0.5	0.9616	0.9612

Table 9: XGBoost Performance Results for Blind Test Dataset

The results listed in table 9 shows the recall, or sensitivity, score, where the ratio of correctly predicted positive observations to the total predicted positive observations, is at over 92% with both train and test samples. This high recall score confirms that of most of the registered voters' party affiliation were accurately predicted the in the XGBoost model. The precision performance measure shows the ratio of correctly predicted positive observations to the total predicted positive observations at over 87%. These scores confirm that even without the most predictive feature, it model is performing with a highly successful accuracy score.

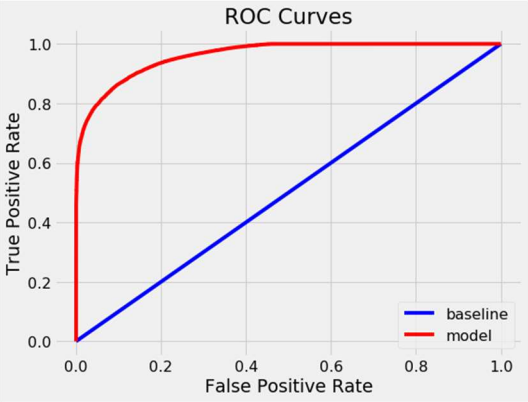


Figure 10 XGBoost ROC Results for Blind Test Dataset

Figure 10 shows the ROC curves for the XGBoost model, which interprets the 96% accuracy scores for both trrain and test samples of the XGBoost model. Again, represented at a point (0,1), figure 10 shows a line travelling from the bottom of the left of the plot and curving to the top right well past the 87% mark. This representation of line similarly shows another high-performing skillful model, where the probability of randomly chosen real positive occurrences versus negative occurrences, is high.

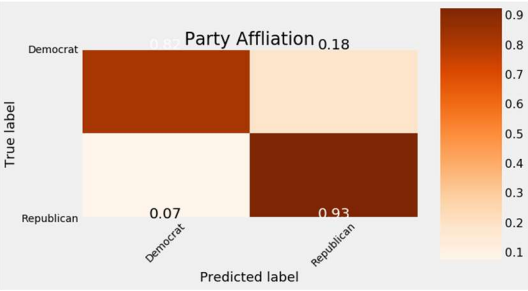


Figure 11: Normalized confusion matrix for XGBoost model for blind test dataset

Figure 11 displays the normalized confusion matrix table describing the performance of the XGBoost classification model, where the 82% of predicted samples were true positives while 18% were false positives; and 92% of the predicted samples were true negatives while 7% were false negatives.

13. conclusion

The objective of this research was to create a base model that predicts a voter's party affiliation. Learning algorithms, including decision tree, random forest and gradient boosting classifiers were utilized towards this objective. Moreover, independent variables such as voter's turnouts and party support in previous elections, residential demographic features, associated congressional district representative's party affiliation and CPVI scores, all had contributing factors in accurately predicting a voter's party affiliation. By applying preprocessing methods, data transformation and reduction led to high-performing and reliable models. Furthermore, a comparison between the learning models (decision tree, random forest and XGBoost) was carried out to identify that the XGBoost is the best model. In conclusion, the comparison demonstrated that the random forest and the XGBoost learning algorithms are very similar, as they have high and near perfect accuracy scores.

However, XGBoost is theoretically a better model overall for a number of reasons, including: (a) Implements regularized boosting to reduce overfitting; (b) implements parallel processing even though it is a sequential process resulting in greater speed; (c) allows for higher optimization through a range of hyperparameter tuning through grid search for evaluation criteria; (d) implements its in-built routine to handle missing values which are important for this type of big data; (e) it splits up till the specified max_depth and then starts tree pruning process backwards to remove splits beyond which there is no positive gain; and finally (f) it runs a cross-validation at each iteration of the boosting process allowing for easily obtaining exact optimum number of boosting iterations in a single run.

The quality of the XGBoost model was also tested by removing the most predictive feature, proving that it continued to perform well with over 90% accuracy.

14. Future Work

Imagine if a PEC appropriated their valuable resources for a marketing campaign that ended up mobilizing voters to cast votes for their opponent. Not only would this counterproductive marketing campaign be an inefficient use of a PEC's funds, but it would also be detrimental to potential voters who may have missed the chance to align themselves with candidates sharing their favored political interests, or even their chance to voice social grievances to PEC candidates in urging positive changes to their society. To curb these potentially flawed campaigns from happening, research as such must continue to find data-driven prediction models with higher accuracies and performance. Utilizing data about voters' socio-political preferences, expected behaviors, responses, past interactions with PECs, and historical record of turnouts at previous elections, is crucial in aligning voters with PECs that share similar socio-political agendas and goals.

The learning model contributed through this research provides a highly accurate model for identifying a voter's registered party affiliation based on a non-exhaustive election turnout history. This contribution has created the foundation for the first version of the prototype for Litics360.

The second step is to develop a secondary predictive model based on the first predictive identification of voter's party affiliation, which assign a specific CPVI number to the voter based on survey answers provided through PEC marketing strategies and polling questions, that is predictive of the voter's likelihood to support their registered party in an election. This secondary model will help identify voters who fall in the middle spectrum of the CPV index allowing for better utilization of PEC marketing and resources for garnering more efficient PEC-to-voter communication strategies.

The third step is to develop the tertiary prediction model to evaluate a voter's turnout probability based on the secondary predictive model's CPVI score, from polled answers, voters' interactions, i.e., page views and visits, for the candidate's landing site, PEC's event attendances, donations, and direct communication with PECs, from the secondary predictive model. This tertiary prediction model will help PECs further, to communicate with the right voters for greater turnout in upcoming elections.

Finally, developing a brand-centric and user-friendly front-end UI/UX, based on the features engineered for prediction models, will integrate interactive methods of visualizing this data in real-time through dashboard UI/UX for both PEC staff and voter users.

15. REFERENCEs

[1] Aldrich, J.H. and McKelvey, R.D. 1977. A Method of Scaling with Applications to the 1968 and 1972 Presidential Elections. *The American Political Science Review*. 71, 1 (Mar. 1977), 111. DOI:https://doi.org/10.2307/1956957.

[2] Alexander, B. et al. *A Bayesian Model for the Prediction of United States Presidential Elections* *.

[3] Ansolabehere, S. et al. 2001. Candidate Positioning in U.S. House Elections. *American Journal of Political Science*. 45, 1 (Jan. 2001), 136. DOI:https://doi.org/10.2307/2669364.

[4] Benoit, K. et al. Wordscores.

[5] Bonica, A. 2016. A data-driven voter guide for U.S. Elections: Adapting quantitative measures of the preferences and priorities of political elites to help voters learn about candidates. *Rsf*. 2, 7 (2016), 11–32. DOI:https://doi.org/10.7758/rsf.2016.2.7.02.

[6] Breiman, L. 2001. Random forests. *Machine Learning*. 45, 1 (2001), 5–32. DOI:https://doi.org/10.1023/A:1010933404324.

[7] Catalist, Y.G. and Gelman, A. 2018. *Voter Registration Databases and MRP Toward the Use of Large Scale Databases in Public Opinion Research*.

[8] Challenor, T. 2017. *Predicting Votes from Census Data*.

[9] Directory of Representatives | House.gov: *https://www.house.gov/representatives#state-ohio*. Accessed: 2020-02-02.

[10] Ecanvasser: 2012. *http://www.ecanvasser.com/*. Accessed: 2019-12-13.

[11] Farrell, D.M. and Webb, P. 2002. Political Parties as Campaign Organizations. *Parties Without Partisans: Political Change in Advanced Industrial Democracies*. (2002). DOI:https://doi.org/10.1093/0199253099.001.0001.

[12] Grimmer, J. and Stewart, B.M. 2013. Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis*. 21, 3 (Jan. 2013), 267–297. DOI:https://doi.org/10.1093/pan/mps028.

[13] Hasan, A. et al. 2018. Machine Learning-Based Sentiment Analysis for Twitter Accounts. *Mathematical and Computational Applications*. 23, 1 (Feb. 2018), 11. DOI:https://doi.org/10.3390/mca23010011.

[14] i360: 2009. *http://www.i-360.com/*. Accessed: 2019-12-13.

[15] Klimeka, P. et al. 2012. Statistical detection of systematic election irregularities. *Proceedings of the National Academy of Sciences of the United States of America*. 109, 41 (Oct. 2012), 16469–16473. DOI:https://doi.org/10.1073/pnas.1210722109.

[16] L2 Political: 1990. *https://l2political.com/*. Accessed: 2019-12-13.

[17] Louwerse, T. and Rosema, M. 2014. The design effects of voting advice applications: Comparing methods of calculating matches. *Acta Politica*. 49, 3 (Jul. 2014), 286–312. DOI:https://doi.org/10.1057/ap.2013.30.

[18] Lowe, W. 2008. Understanding Wordscores. *Political Analysis*. 4, (2008), 356–371. DOI:https://doi.org/10.1093/pan/mpn004.

[19] Matsusaka, J.G. and Palda, F. Voter Turnout: How Much Can We Explain? *Public Choice*. Springer.

[20] Mian, A. and Rosenthal, H. 2016. Introduction: Big data in political economy. *Rsf*. Russell Sage Foundation.

[21] NationBuilder: 2009. *https://nationbuilder.com/*. Accessed: 2019-12-13.

[22] Nickerson, D.W. and Rogers, T. 2014. Political campaigns and big data. *Journal of Economic Perspectives*. 28, 2 (2014), 51–74. DOI:https://doi.org/10.1257/jep.28.2.51.

[23] Peress, M. 2013. Estimating proposal and status Quo locations using voting and cosponsorship data. *Journal of Politics*. 75, 3 (Jul. 2013), 613–631. DOI:https://doi.org/10.1017/S0022381613000571.

[24] Poole, K.T. and Rosenthal, H. 1985. A Spatial Model for Legislative Roll Call Analysis. *American Journal of Political Science*. 29, 2 (May 1985), 357. DOI:https://doi.org/10.2307/2111172.

[25] Raileanu, L.E. and Stoffel, K. 2004. *Theoretical comparison between the Gini Index and Information Gain criteria* *. Kluwer Academic Publishers.

[26] Smith, S. et al. 2012. Predicting Congressional Votes Based on Campaign Finance Data. (2012). DOI:https://doi.org/10.1109/ICMLA.2012.119.

[27] State Laws on Access to and Use of Voter Registration Lists: 2019. *http://www.ncsl.org/research/elections-and-campaigns/access-to-and-use-of-voter-registration-lists.aspx*. Accessed: 2019-12-01.

[28] Statewide Voter Files Download Page: *https://www6.ohiosos.gov/ords/f?p=VOTERFTP:STWD::#stwdVtrFiles*. Accessed: 2020-04-18.

[29] TheOptimizer: 2016. *https://theoptimizer.io/*. Accessed: 2019-12-13.

[30] Tune Learning Rate for Gradient Boosting with XGBoost in Python: *https://machinelearningmastery.com/tune-learning-rate-for-gradient-boosting-with-xgboost-in-python/*. Accessed: 2020-04-23.

[31] US Constitution 1791. First Amendment Religion and Expression. United States Senate, Office of the Secretary of the Senate, Library of Congress.

[32] Wolpert, D.H. and Macready, W.G. 1997. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*. 1, 1 (1997), 67–82. DOI:https://doi.org/10.1109/4235.585893.

16. acknowledgements

I would like to give a special thanks to Dr. Afzal Upal, Chair of Department of Computing and Information Science at Mercyhurst University, for his valuable guidance during my research. I would also like to acknowledge and give thanks to data scientist, Shraddha Dubey, for always being a wonderful sounding board during the critical brainstorming sessions of this research.

About the author:

Sabiha Mahmud Sumi is a graduate student at Mercyhurst University in Erie, PA. She is a multidisciplinary designer specializing in helping businesses strategize in bridging the gap between business narratives, branded design, impactful marketing and data-driven learning. She has a Bachelor of Arts in Graphic Design from Mercyhurst University. Previously, she was a journalist at several leading newspapers in Dhaka, Bangladesh. She was also the owner and creative director for two restaurants in Dhaka, Bangladesh. For more information about the author, please visit: https://mahmudsumi.com/resume.
For more details about Litics360, please visit: https://github.com/mahmudsumi/litics360

Predicting Enrollment Behavior of Traditional Freshmen at Mercyhurst University

Kyndra Zacherl
Department of Computing &
Information Science
Mercyhurst University
501 East 38th Street
Erie, PA 16546
kzache61@lakers.mercyhurst.edu

ABSTRACT
In the competitive world of higher education, enrollment managers must seek novel solutions and strategies in order to maximize the number of students who enroll each recruitment cycle. This paper examines three years of Mercyhurst University accepted student data (excluding athletic recruits, international students, and students with missing/bad CEEB codes), alongside socioeconomic features retrieved from the U.S. Census Bureau and the National Center for Education Statistics, to determine the likelihood of each applicant's enrollment at the institution. A logistic regression model and three classifiers— decision tree, random forest, and support vector machines (SVMs)—were utilized in order to predict an applicant's likelihood of enrollment, measured by an enrollment efficiency score (a metric that considers all outcomes other than false negatives to be a success). The average enrollment efficiency score from 10 iterations of each model were compared in order to select a final model, SVMs. The model's enrollment predictions will be provided to Division of Enrollment personnel to aid in recruitment efforts with the goal of reallocating resources towards applicants who are predicted to enroll but have not yet done so.

KEYWORDS
Enrollment management, higher education, logistic regression, random forest, decision tree, support vector machines, machine learning

17. INTRODUCTION
Enrollment management is a structural concept that emerged in the latter half of the 20th century, consisting of a variety of marketing campaigns, financial aid policies, admissions techniques, organizational strategies, and institutional research specifically designed to optimize the revenue of an institution and achieve the most advantageous level of student enrollment [8]. However, the current state of college enrollment in the United States is dismal; 2019 has marked the eighth consecutive year that nationwide enrollment has declined, with Pennsylvania being one of the top five states affected by this slump [1]. Yield, defined as the proportion of accepted students that enroll at an institution, has fallen by almost 13 percent from 2002 to 2019 at private four-year colleges, with the current average student yield being 33.4 percent [4, 10]. This indicates that of all applicants that a college accepts, only a third confirm their enrollment and attend. While some highly selective universities may be suffering from overenrollment, the large majority of universities are plagued by underenrollment (as well as limited resources), giving enrollment managers the paradoxical goal of expending finite resources to maximize enrollment, best done by attempting to increase institutional yield [5].

Predictive modeling has emerged as a potential ally for enrollment managers seeking novel solutions to effectively recruit and enroll students. A successful use case of this method is the University of Oklahoma, which utilized modeling to increase both the overall yield and academic quality of students. In this case, data scientists used four models (decision trees, logistic regression, forward stepwise regression, and backward stepwise regression) to predict a student's probability of enrolling, with separate models for students who indicated on-campus and off-campus residency preferences. These models resulted in 88-92 percent accuracy in forecasting whether a student would enroll at the institution, with admissions representatives relying on visual interpretations of the model to tailor financial aid packages to students that were deemed more likely to enroll [6].

This paper aims to develop a model for predicting traditional freshman enrollment at Mercyhurst University. Although the University of Oklahoma was extremely successful in their effort to increase enrollment numbers with predictive modeling, a variety of adjustments need to be made to their modeling to better fit Mercyhurst University. The primary modification is to exclude transfer students, athletes, and international students from the dataset and only examine the behavior of traditional freshmen. The aforementioned student types do not display typical behavior in the enrollment process and would not contribute to accurate attempts to predict a traditional student's decisions. For example, the cost of recruiting a traditional undergraduate at a private institution in 2018 was $2,357, while an international student cost $735 and a transfer student only cost $302 [16]. Athletes are also outliers due to the NCAA recruiting process and the various stipulations attached to their recruitment. Additionally, the University of Oklahoma only utilized the prediction of their model to adjust financial aid packages. In the context of Mercyhurst University, the model will be utilized to allocate resources (e.g. counselor phone calls and texts, mailings, T-shirts) towards the accepted students and students who have started but not submitted their applications and are most likely to enroll. The University of Oklahoma also notes that "most" of their variables were "unreliable, missing, or incomplete," which is not ideal when attempting predictive modeling [6].

Although previous research in the field has attempted to discern aspects associated with student enrollment behavior, this model combines variables from three highly reliable sources (confidential

Mercyhurst University enrollment data, NCES data, and U.S. Census Bureau data) and attempts to predict whether a student will enroll, while only considering the most "predictable" student type, traditional freshmen. With the guidance of this model, the goal of the Mercyhurst University Division of Enrollment is to increase overall student yield for Fall 2020 by allocating more resources to prospective students that are predicted to enroll but have not done so yet. The final assessment for this model will involve using it to predict the likelihood of enrollment for the hundreds of started-but-unsubmitted applications for the 2020 recruitment cycle, based on the minimal data they have provided the institution, should they complete their applications and be accepted.

With a nationwide stagnation in college enrollment, it is imperative for an institution to optimize enrollment in order to ensure future financial viability. Proper resource allocation is key—devoting valuable time and money to a student that has a very low likelihood of attendance is of no benefit and having this knowledge early in the recruitment process is of the utmost importance. Although this model is specifically developed for Mercyhurst University enrollment data, the underlying assumption regarding the behavior of traditional freshmen will remain.

18. RELEVANT WORK

A number of higher education studies have utilized data science methods, but only a select few relate specifically to machine learning and enrollment. However, none of them precisely cover the topic of predicting enrollment by examining a prospective student's socioeconomic and demographic data.

Jia and Mareboyana [3] examined the effectiveness of different machine learning algorithms to predict student retention at historically black colleges. Their study compared the classification accuracy of neural networks, WEKA decision trees, and SVMs as classification methods. Their results showed that all three models produced high accuracy levels (90+ percent), but SVMs ultimately showed the highest level of prediction accuracy. However, Jia and Mareboyana's data set consisted of only 771 students—a much more limited set than the accepted students dataset used in this paper. Additionally, their work only focused on historically black colleges, whose students may display behavior that is not consistent with that of applicants to other types of institutions.

Slim et al. [21] utilized logistic regression, SVMs, and semi-supervised probability methods to predict an applicant's probability of enrolling at the University of New Mexico. This method utilized two approaches to classification: cohort-level classification and individual-level classification. Given a set of features, cohort-level classification predicted the portion of all prospective students that can be expected to enroll while individual-level classification returned a 0 or 1 flag as to whether a specified student will enroll. The results of this process showed that certain attributes (financial award, timing of admissions decision, GPA, state of residency) are highly correlated to enrollment, while many other features show little correlation or are completely irrelevant. Although Slim et al.'s report has the same goal as that of this paper (maximizing university enrollment), primarily using logistic regression limited the capabilities of the classification, as it could only identify non-linear patterns in the data. While Slim et al.'s paper examined a large number of features, the variables were only those in the university database, and did not include U.S. Census Bureau or NCES data as did this paper. Furthermore, the university of interest in this case is a public school, and there are noted differences in enrollment behavior between public and private universities [10], meaning their conclusions may not be applicable to Mercyhurst University.

In Ragab et al. [14], the authors compared machine learning algorithms to classify student enrollment, selecting nine from the WEKA library: C4.5, Random Forest, IBK, IBK-E, IBK-M, LIBSVM, MLP, multilayer perceptron, and PART. Through the use of ten-fold cross validation to evaluate the accuracy of each method, each student was classified into the college that they were most probable to enroll in (medical, engineering, etc.). This paper determined that C4.5 gave the best overall performance, followed closely by PART, Random Forest, multilayer perceptron, and MLP. Although this work tested a variety of algorithms, it is extremely limited on the details of what data and features were included to draw these conclusions. Additionally, only algorithms that can be run in WEKA were considered for testing, which limited the methods that could be considered.

Nandeshwar and Chaudhari [7] investigated the use of decision trees and rules to predict whether a student would "survive" (i.e., graduate) from West Virginia University. This dataset consisted of 248 features (including a variety pulled from the U.S. Census Bureau) and 28,000 applications over the course of seven years. The authors created a number of flags (e.g., first-generation college student, large amounts of financial aid received) to accompany the data and removed all students that were not accepted. Upon examination of the variables, it was determined that financial aid was ultimately the greatest indicator of whether a student enrolled, but not whether they remained enrolled. This study has a number of limitations, particularly the fact that ultimately the authors concluded that only one of the many features was worthwhile in pursuing, and that single feature was ultimately not associated with the premise of their study (retention). In making this conclusion, they suggested that financial aid be used to control the quality of students, suggesting an overenrollment problem, whereas the goal of this paper is to maximize enrollment.

19. PROPOSED SOLUTION

As no existing model solves the problem of predicting enrollment at Mercyhurst University, this paper proposes a machine learning approach that predicts the likelihood of an applicant enrolling when provided their application data and basic socioeconomic and geographic features. These predictions will then be utilized by Division of Enrollment personnel to focus on applicants who the model predicts will enroll (but have not done so yet) and allocate resources towards them appropriately.

The dataset for this project originates from three years of Mercyhurst University applications from Fall 2016, Fall 2017, and Fall 2018. This dataset intentionally includes only students who applied for fall entry, as this is the semester in which traditional freshmen first enroll. The final dataset excludes students who were not offered admission to the university (as their outcomes are irrelevant to yield), and students that applied as athletes, transfers, or international students, due to their atypical enrollment behavior. Additional manual exclusions were performed to eliminate applications with missing/bad data.

In addition to the data provided by a student's application, more information can be obtained from the U.S. Census Bureau and NCES. However, the Division of Enrollment has requested that this project only utilize features that can be obtained with the basic information found in started-but-unsubmitted applications, as that is the data source the selected model will eventually be deployed on. Therefore, this project focused on two key variables to gather

further data: an applicant's five-digit home ZIP code and one or more CEEB codes that identify the high school(s) of an applicant. Relevant socioeconomic data for each unique ZIP code was obtained from the U.S. Census Bureau, consisting of variables that represent four measures: marriage rate, home ownership rate, poverty level, and educational attainment.

The accepted student data, now joined with all gathered features, was passed into a logistic regression model, which then attempted to predict the likelihood of a student enrolling. This logistic regression model provides a baseline to compare the performance of three classifiers (SVMs, decision trees, and random forest) at successfully predicting student enrollment.

Performance will be evaluated by dividing the sum of the true positives, true negatives, and false positives by the total number of applications in the test data and then comparing these values for each model; this metric will be referred to as the "enrollment efficiency score." This evaluation metric was chosen because false negatives (applicants who the model predicts will not enroll who actually would have) are very detrimental to enrollment and must be avoided at all costs. Therefore, the enrollment efficiency score counts all outcomes that are not false negatives as a success. False positives (applicants who the model predicts will enroll who actually would not) are also undesirable and can exhaust resources but are expected in recruitment and do not significantly negatively affect the process.

The phases of this project can be summarized as follows:

1. Obtain accepted student data, consisting of (at minimum) an applicant's home ZIP code, high school CEEB code, and enrollment status
2. Gather additional features from the U.S. Census Bureau using ZIP code and from NCES (matched to each application using their CEEB code)
3. Train a logistic regression model and evaluate its performance on the test data by calculating the enrollment efficiency score
4. Train three classifiers and evaluate their performance on the test data by calculating their enrollment efficiency score
5. Compare the enrollment efficiency score of the four models to select a final model which will be deployed on existing started-but-unsubmitted applications

20. METHODOLOGY

The initial accepted students dataset of 7,653 applications consists of the following columns: entry term, application ID, street address, admissions region (e.g., Western Pennsylvania), enrollment status (the target variable), origin code/description (e.g., college fair), and CEEBs codes (up to five that identify previously attended high schools or colleges). As the dataset was pre-filtered to only include traditional freshmen for fall applications, no adjustments needed to be made to exclude transfers or adult student applications. However, a variety of adjustments needed to be made to prepare the original dataset to be matched with additional features (and eventually, for modeling). All rows with a missing ZIP code were removed, as merging the U.S. Census Bureau features relies on ZIP code matching. Any application with a ZIP code not recognized as a U.S. Census Bureau ZIP Code Tabulation Area (ZCTA) [22] was removed—these mostly consisted of applications with a PO Box in a non-residential area. To exclude international applications, any application with a zone of "INTERNATIONAL," a non-U.S. ZIP code (including Guam and Puerto Rico), or with a CEEB code for a foreign institution (namely, 777777) were removed. All applications with a zone relating to athletic recruiting (e.g., "BASEBALL") were removed. Any application with a missing CEEB code was removed; for those with a non-existent first CEEB code, their subsequent CEEB codes were examined, and if those were also missing or non-existent, their application was removed. After these eliminations, the final accepted students dataset consisted of 6,308 applications.

To retrieve data from the U.S. Census Bureau, an API key was obtained and utilized to request eight variables from the 2014-2018 5-Year American Community Survey [23] for each unique ZCTA in the accepted student dataset. These variables, selected to represent a variety of socioeconomical measures, consisted of: persons aged 25+ with a bachelor's degree (B15003_022E), total persons aged 25+ (B15003_001E), households below the poverty level in the past 12 months (B17001_002E), total households (B17001_001E), owner-occupied residences (B25003_002E), total occupied residences (B25003_001E), married residents (B06008_003E), and total residents (B06008_001E). To determine the ratio of each measure for each ZCTA, the variable of interest was divided by its associated total variable. The Fisher-Jenks algorithm [15] was then used to locate natural data breaks within each measure—minimizing variance within the groups and maximizing it between the groups—and assign all observations to one of four buckets represented by categorical variables: "least," "less," "more," or "most." These measures were then matched on each application by ZIP code.

The vast majority of U.S. educational institutions, both public and private, submit institutional data to NCES, which is accessible via their Elementary/Secondary Information System (ELSi) [9]. For public schools, the NCES ID, street address, locale (rural, town, suburb, or city), number of students eligible for free or reduced-price lunch (FRPL), and total number of enrolled students were retrieved. For private schools, the NCES ID, street address, and locale were retrieved (the national FRPL lunch program is only offered in publicly funded schools). To determine the percentage of students who are FRPL-eligible in each public school, the number of FRPL-eligible students was divided by the total number of students enrolled. In accordance with national FRPL program guidelines [12], each public school was then classified as a school type of "public, low-poverty," "public, mid-low poverty," "public, mid-high poverty," or "public, high-poverty" based on their percent of FRPL-eligible students; all private schools were classified with a school type of "private," including homeschools (CEEB code of 970000).

Because NCES identifies high schools with an NCES ID and the accepted student dataset identifies high schools with CEEB codes, an NCES ID-CEEB code crosswalk was created in order to match the data gathered from NCES to each application in the accepted student dataset. For each high school exported from NCES, a unique string was generated consisting of the numbers in its street address, its ZIP code, and its city, e.g., "210085364YUMA." This string was also generated for a list of high school street addresses and their respective CEEB codes, provided by the Division of Enrollment. These strings were then matched to create this crosswalk, which allowed the NCES features to match to the CEEB codes in the accepted student dataset. Any high schools within the accepted student dataset that did not have an NCES ID-CEEB code match were matched manually.

Two additional features were generated that allowed for an examination of the distance from Mercyhurst University to the ZIP

code provided on an application and the number of other applicants at the high school identified in an application's CEEB code. A list of ZIP codes and their respective latitudes and longitudes was provided by the Division of Enrollment, and the number of miles between each ZIP code and Mercyhurst University was calculated using Haversine distance [13]. The Fisher-Jenks algorithm was again applied to determine natural breaks within this measure, and a value of "least," "less," "more," or "most" was assigned to each ZIP code (in respect to distance), which was then joined to the accepted student dataset. A list of CEEB codes for "feeder high schools," defined as high schools with at least 20 applications in the past two years (a total of 40), was generated from the Mercyhurst University enrollment management system. Applications which had a primary CEEB code on this list were valued as a "yes" for a new "feeder_status" variable; all other applications were valued as a "no."

The application origin variables were grouped with those of similar scope to create new categories: applications that originated from ACT or SAT scores were put into a "test scores" category, applications that originated from campus tours or admissions events were put into a "campus event" category, etc. A total of nine distinct categories were created (Learning Differences/Autism Initiative at Mercyhurst, campus event, counselor contact, inquiry, list buy, Mercyhurst application, The Common Application, personal referral, test scores). All other application origin codes (including those that were blank) were assigned to an "other" category. The enrollment status for each application was examined and assigned a 0 or 1 based on whether the student withdrew their application or enrolled at the institution, respectively.

The final accepted student dataset consisted of features representing entry term, admissions region, application origin, ZIP education level, ZIP poverty level, ZIP home ownership rate, ZIP marriage rate, high school locale, high school type, ZIP distance level, and high school feeder status, alongside the target variable of enrollment status. The final set of features was converted to k-1 dummy variables [24] for each category to avoid collinearity and improve computing performance, with the reference group being the first column of each category (setting the "drop_first" parameter to "True" in Pandas' get_dummies function [11] accomplishes this).

A logistic regression model [2] was first generated to provide a baseline enrollment efficiency score for each of the classification algorithms to be compared to. The accepted student dataset was equally split into training and test datasets, with 3,154 applications being placed in each. The logistic regression model, set to a maximum number of iterations of 3,000 and a solver of "lbfgs," was run 10 times on random combinations of training and test data, and the enrollment efficiency score was noted.

A decision tree model [2] was generated from the same dataset and applied to both the training and test data (with the same .5 split). The parameters of the decision tree were tuned by comparing the enrollment efficiency score for 100 iterations for each combination of maximum depth (one through 25) and split criterion ("gini" or "entropy") with 10-fold cross-validation on the training data; as the most successful combination of parameters was Gini Impurity and a maximum depth of 22, these parameters were selected for the final decision tree. This parameter combination was also applied to the random forest classifier [2], in addition to the parameters of a random state of 0 and the number of trees set to 1000. Both the decision tree and random forest models were run 10 times on random combinations of training and test data, and the average enrollment efficiency score for each was noted.

The kernel for the SVMs model [2] was selected by comparing the enrollment efficiency score for 100 iterations for each kernel of interest (linear, polynomial, radial basis function, and sigmoid) and selecting the one with the best average performance on 10-fold cross-validation on the training data—sigmoid. The SVMs model with a kernel of sigmoid were then run 10 times on random combinations of training and test data, and the average enrollment efficiency score was noted.

21. RESULTS AND DISCUSSION

After 10 runs of the logistic regression model, which serves as the baseline for evaluating the success of the three classification algorithms, the average enrollment efficiency score was found to be 0.8047, which can be interpreted as the model predicting a true positive, true negative, or false positive in 80.47 percent of instances in the test data. The features that had the greatest effect on this model (determined by examining the "coef_" attribute [18] of the model) were found to be applications with an admissions region of Learning Differences/Autism Initiative at Mercyhurst program, applications with an admissions region of Music, and applications with an origin of "Other."

The decision tree classifier returned an average enrollment efficiency score after 10 iterations of 0.8328. The most relevant features for the final decision tree (determined by examining the "feature_importances_" attribute of the model [20]) were found to be applications whose origin was The Common Application, applications with an admissions region of Erie County (i.e., local students), applications with an admissions region of Music, and applications with an admissions region of Eastern Pennsylvania/New Jersey.

The random forest model, tuned using the same basic parameters that were found to optimize the decision tree, was found to have an average enrollment efficiency score of 0.8108 after 10 iterations. The features that had the most effect on the random forest classifier (determined by examining the "feature_importances_" attribute [17] of the model) were found to be applications whose origin was The Common Application, applications with an admissions region of Erie County, applications with an origin of "other," and applications with a distance level of "less distance."

The SVMs model was found to have an average enrollment efficiency score of 0.8354 after 10 iterations. Feature weights (coefficients) are only assigned in SVMs using a linear kernel [19], and because sigmoid was selected as the kernel for this model, feature importance could not be determined. The confusion matrix for this model can be found below in Figure 1 (the enrollment efficiency score for this particular model can be calculated by Figure 1 and was found to be 0.8424).

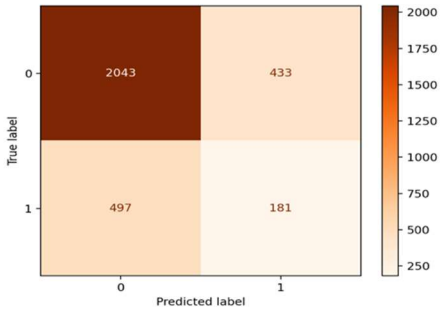


Figure 6: Confusion matrix for the SVMs model

Overall, the SVMs model was found to be the most successful model, with the enrollment status of students being correctly predicted 77.81 percent of the time. This result mirrors the outcome of the analysis undertaken by Jia and Mareboyana [3], who found SVMs to have the highest level of accuracy at predicting enrollment when compared to decision trees and neural networks. In comparison to the enrollment efficiency score of the baseline logistic regression model, all three of the models that utilized classification algorithms showed better performance, as was predicted. The overall range of average enrollment efficiency score for all four models examined in this paper was 80.47-83.54 percent.

22. CONCLUSIONS AND FUTURE WORK

Although the logistic regression model has a number of limitations compared to the classification algorithms, it was found to have a minimal performance difference compared to the worst performing classification algorithm—the random forest model only provided a 0.61 increase in enrollment efficiency score. However, when applied to the thousands of applications that are seen each recruitment cycle, this .61% performance boost would correlate to more than 100 applications not having their enrollment status predicted as a false negative, so utilizing a classification algorithm, despite the tradeoff in complexity and computing power, is only logical. Since the SVMs model provided the best average enrollment efficiency score of all four models, SVMs were selected as the final model for this paper.

The use of a custom metric to score model performance which only penalizes for false negatives was key to evaluating these models for enrollment use, as false negatives represent the most grievous type of error (not dedicating resources towards a student who actually would have enrolled). The traditional scoring metrics of Receiver Operating Characteristic (ROC) and Area Under the ROC Curve (AUC) would have penalized the model for both false negatives and false positives, which are not a large concern in enrollment efforts. As can be seen in Figure 2, the AUC for the logistic regression model was found to be 0.69, while the enrollment efficiency score, which can be calculated from the confusion matrix in Figure 3, was found to be 0.8066.

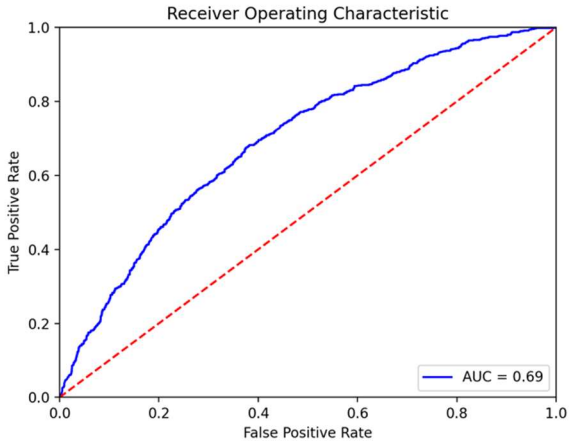


Figure 7: ROC Curve and AOC for the logistic regression model

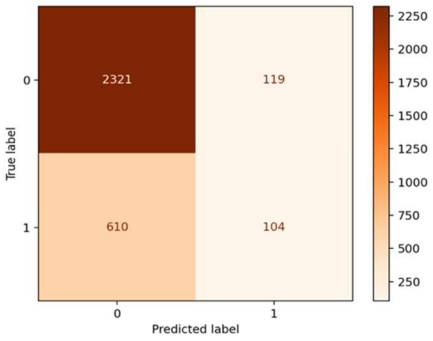


Figure 8: Confusion matrix for the logistic regression model

As was discovered in the logistic regression and SVMs models, applications with admissions region features associated with both the Learning Differences/Autism Initiative at Mercyhurst program and the Music program were found to be highly correlated. In future work, these two regions could be excluded from the models (similar to applications who were identified as athletics recruits or transfers), as it appears that applications who applied under these programs display atypical enrollment behavior when compared to traditional freshmen. This would assist in fine-tuning the model to examine the behavior of the average student, and not one whose acceptance into a special program could affect the likelihood of their enrollment.

The random forest model determined that applications with an origin of "other" were highly correlated. This "other" category served as a catch-all for applications whose origin did not fit with the other nine distinct categories and those that were blank (no specific source for the application). As this application origin category of "other" was important to the random forest model's predictions, it would be worthwhile to investigate the codes contained within the "other" category, and possibly create new categories based on any discernible patterns or assign application origins to those that are currently blank.

Ultimately, the goal of this model would be to provide Division of Enrollment personnel with a visual interpretation of the model on

an application level, allowing them to target individual applications with more resources. At present, this model does not do so, and only provides output of the predicted values in Python, which would then need to be interpreted for non-technical users. An improvement upon this project would be to create an export that lists applications that were predicted to enroll and have not done so yet, sorted by admissions region and emailed each week, so that each admissions counselor automatically receives a list of their most valuable applications and can allocate resources appropriately.

The SVMs classifier that was created for this paper will be deployed on all accepted students who have not yet enrolled and all started-but-unsubmitted applications for the current recruiting cycle (Fall 2020), and the results will be manually interpreted and distributed to the appropriate Division of Enrollment personnel for resource allocation towards the applications that the model predicts will eventually enroll.

23. REFERENCES

[1] Fain, P. 2019. College Enrollment Declines Continue. *Inside Higher Ed*.

[2] Hackeling, G. 2017. *Mastering Machine Learning with Scikit-Learn*.

[3] Jia, J.W. and Mareboyana, M. 2013. Predictive models for undergraduate student retention using machine learning algorithms. *Proceedings of the World Congress on Engineering and Computer Science*. I, (2013), 315–329. DOI:https://doi.org/10.1007/978-94-017-9115-1_24.

[4] KelmscottEDU 2017. *Essential Strategies to Increase Your Yield Rates*.

[5] King, A. 2019. Improving the Enrollment Process through Machine Learning. *EDUCAUSE Review*.

[6] Mariani, G. 2017. How one university reached a new enrollment record with 4 predictive analytics models. *eCampusNews*.

[7] Nandeshwar, A. and Chaudhari, S. 2009. Enrollment prediction models using data mining. (2009).

[8] National Association of Independent Schools 2012. *What is Enrollment Management?*

[9] National Center for Education Statistics 2019. Elementary/Secondary Information System (ELSi). (2019).

[10] National Student Clearinghouse Research Center 2019. *Current Term Enrollment – Spring 2019*.

[11] Pandas 2014. pandas.get_dummies. (2014).

[12] Public school students eligible for free or reduced-price lunch: 2019. *https://nces.ed.gov/fastfacts/display.asp?id=898*.

[13] PyPI 2020. Haversine. (2020).

[14] Ragab, A.H.M. et al. 2014. A comparative analysis of classification algorithms for students college enrollment approval using data mining. *ACM International Conference Proceeding Series* (2014), 106–113.

[15] Rey, S.J. et al. 2017. An evaluation of sampling and full enumeration strategies for Fisher Jenks classification in big data settings. *Transactions in GIS*. 21, 4 (2017), 796–810. DOI:https://doi.org/10.1111/tgis.12236.

[16] Ruffalo Noel Levitz 2018. *2018 Cost of Recruiting an Undergraduate Student Report*.

[17] Scikit-learn sklearn.ensemble.RandomForestClassifier.

[18] Scikit-learn sklearn.linear_model.LogisticRegression.

[19] Scikit-learn sklearn.svm.SVC.

[20] Scikit-learn sklearn.tree.DecisionTreeClassifier.

[21] Slim, A. et al. 2018. Predicting student enrollment based on student and college characteristics. *Proceedings of the 11th International Conference on Educational Data Mining, EDM 2018* (2018), 383–389.

[22] United States Census Bureau *ZIP Code Tabulation Areas (ZCTAs)*.

[23] Variables in American Community Survey 5-Year (2014-2018): 2018. *https://api.census.gov/data/2018/acs/acs5/variables.html*.

[24] Wright, M. and König, I. 2019. Splitting on categorical predictors in random forests. *PeerJ*. 7, e6339 (2019). DOI:https://doi.org/10.7717/peerj.6339.

Student Retention Analysis

Udip Bohara
Department of Computing & Information Science
Mercyhurst University
Erie, Pennsylvania
ubohar00@lakers.mercyhurst.edu

ABSTRACT
Students are primary assets of any educational entity. Hence student retention is a salient predictor of success in the educational field. This paper utilizes machine learning techniques to analyze student retention. Logistic Regression, Decision Trees, Support Vector Classifier and Random Forest Classifiers through features engineering are utilized to build a classification model that predicts student retention. In order to build sound models, imbalanced data is handled in two different methods: random under sampling and synthetic over sampling. This is done to preserve the sensitivity of the model. Important features derived from these models are then utilized to highlight salient features that have high predictability towards student retention which can be utilized for intervention strategies. Ultimately, this paper ranks 'at risk' students with probabilities derived from Logistic Regression models.

Keywords
Student Retention, Churn Analysis, Machine Learning, Features Engineering, Imbalanced Data, SMOTE, Classification

1. INTRODUCTION
In fall 2018, the full-time retention rate in postsecondary institutions was 75.5% [1]. This is based on 5,135 institutions. Graduates are the product of a university. In order to have a high yield of production, it is important to continuously assess and strengthen the factors that drive production. For this domain, student retention is the most salient factor that dictates production. Therefore, identifying the key factors affecting retention is necessary to apply intervention techniques to decrease the number of students who cease enrollment each year.

Dr. Lau explored the institutional factors affecting student retention [2] and found appropriate funding, academic support services, and the availability of physical facilities, in addition to the effective management of multiculturalism and diversity on campus to be the important factors that played a significant role in retaining student churn. According to Seidman, Early identification along with intensive and continuous intervention is necessary for successful retention [3].

$$Ret = Early_{Identification} + (Early + Intensive + Continuous)_{Intervention} \quad (1)$$

Traditionally, many universities have used academic performance indicators such as GPA and standardized test scores (SAT and ACT) of students to generate rules that can be used to identify at-risk students [4]. Thus, one way to increase retention is to increase academic performance. However, from research done in the past [2,3,4], student retention has been found to be influenced by a multitude of factors. Therefore, an effective model cannot be limited in the features highlighted, as this will result in a faulty intervention strategy that fails to include all the factors that affect student retention. For example, building a model that only highlights either academic or social-economic features would be erroneous because both play an important role in student retention. Also, it is important to highlight features that are relevant to the university at the time of the features being assessed and analyzed. High-school academic performance along with standard test scores should be in the domain of student admission rather than retention. Student admission measures and criteria such as standards for acceptance are beyond the scope of this study.

In the past, predictive modeling techniques have been utilized to analyze student retention [5,6,7,8]. Classification modeling through features engineering and domain knowledge are used to build an intervention product that is successful in retaining students. This paper focuses on analyzing student retention among students who have at least completed a semester of study. It deviates from the prevalent norm of analyzing only the First-Year students (traditional freshmen) [5,6,7,8] by including students of all academic levels. In doing so, most secondary education data such has High School GPA and other metrics are purposefully ignored. Necessary modeling techniques are utilized for unbalanced dataset to ensure preservation of sensitivity of the model. A sound model for this project should have high sensitivity as it should correctly identify students who cease enrollment.

2. RELEVANT WORK
24. Conventionally, various classification models have been applied to tackle the issue of student retention with machine learning. Alkhasawneh utilized a hybrid neural network to predict student first year retention [5]. This research used feed forward back-propagation architecture for modeling. Each model was built in two different ways: the first was built using all available student inputs, and the second using an optimized subset of student inputs. The optimized subset of the most relevant features that comes with the student, such as demographic attributes, high school

rank, and SAT test scores was formed using genetic algorithms. Whitlock used Principle Component Analysis for dimensionality reduction to build predictive models using classification algorithms [6]: logistic regression, decision trees, neural networks and support vector machines. Statistical tests such as correlations and dimensionality reduction measures are fundamental part of features engineering that helped build a model free of issues such as collinearity. Murtaugh et al proposed a multivariate and univariate churn analysis of freshman students [7] where smaller number of features were considered for retention analysis. Survival analysis was done which allowed the efficient modeling of student retention based on data from recently enrolled students as well as from students who have already graduated or dropped out. Rajuladevi proposed the use of classification methods to identify at-risk students at an early stage, so that the interventions could be offered in a timely manner for first year students [8] through drawing out probabilities from logistic regression. Intervention techniques inferred from these algorithms are beneficial towards building a sound pragmatic system.

25. Chawla et al, proposed SMOTE (Synthetic Minority Over-sampling Technique) to handle imbalances in data. SMOTE works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line [9]. Specifically, a random example from the minority class is first chosen. Then k of the nearest neighbors for that example are found (typically k=5) as shown in Figure 1. A randomly selected neighbor is chosen, and a synthetic example is created at a randomly selected point between the two examples in feature space.

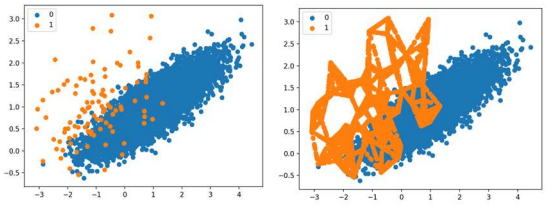


Figure 1: A visual example of how SMOTE algorithm is utilized. Left is the original generic dataset with minority class of '1' which is synthetically oversampled on the right to create a balanced dataset.

Data is imbalanced when number of instances in one class is much smaller than the number of instances in other class. Therefore, during the training stage, classifiers take more sample from the classes which have bigger number of instances [10]. Due to this, at test stage, classifiers are less sensitive to the classes which have smaller number of instances. Jia et al, used SMOTE in pre-processing to solve data imbalanced problem in building classification algorithms such as Decision tree, Support Vector Machines (SVM), and neural networks supported by Weka software toolkit [11].

3. PROPOSED SOLUTION

This paper deviates from First-Year student analysis and focuses on retention among students who have at least completed a semester. It also includes students of all class levels: freshmen, sophomores, juniors and seniors so that analysis for the entire student population can be done. By utilizing a dataset that is frozen after a completion of an academic year 2017, it ensures that the students have completed at least a semester of study at the university. The reason to do so is to use the model as a tool to assess students who have had a substantial presence on campus. Hence, this solution attempts on ignoring data prior to post-secondary education. Features modeling and engineering is the important to generate a robust dataset which can be used for further predictions. Features are selected and imputed situationally to maintain the authenticity of the data. In doing so, features with substantial missing values is disregarded. Unbalanced dataset is handled through two different methods: SMOTE and random under sampling. The ultimate goal of this project is to identify 'at-risk' student of ceasing enrollment.

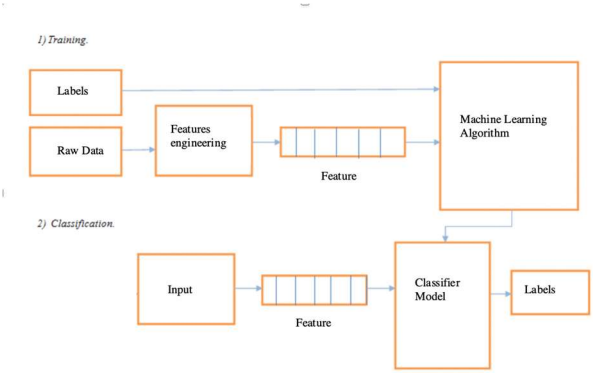


Figure 2: Phases of the project

4. DATA COLLECTION

The data for this research was collected through Ellucian Colleague which utilizes integration model to bring the institution's data together. Multiple files were extracted and then combined to build a dataset. The primary data consisted of students from the year 2008 to mid 2019. Secondary data consisted students who ceased enrollment for the corresponding years and consisted of 11191 unique cases of ceases.

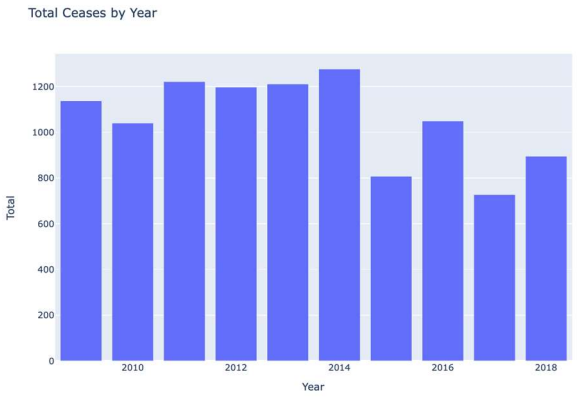


Figure 3: Total ceases by year

4.1 Data Preparation

Features engineering was essential to draw out relevant information from the data. The primary purpose of the extracted data was storage rather than analysis. The data underwent a lot of change over the years due to various factors such as changes in academic calendar system (trimesters to semester system), academic reforms (revamp of curricular system), change in methods of data storage, change in data storage methods (codes for features) and manual errors in data entry (missing and/or erroneous records). Due to above inconsistencies, the primary data consisted of a lot of noise. Hence, it was essential to filter the data for modeling purposes.

For the purposes of retaining integrity, relevance and practicality of the data, data from the year 2017 was selected to be the primary source of the data to build the dataset for modeling. The dataset consisted of all students who were fulltime students at the end of the academic year (May) 2017. Graduate students were not included in the study as the study pertained to analyze retention among undergraduates. The students who ceased enrollment the next year were identified through cease reports of 2018 and binary classifiers (labels) were created:

a) Students who ceased enrollment in 2018 as 1

b) Students who retained their academic status the next year as 0.

Since the dataset consisted of categorical as well as quantitative features, univariate chi square analysis was done to ensure that collinearity was reduced. Multicollinearity has several negative side-effects. It makes it difficult to determine the individual effect of a predictor because the predictors are correlated. Any missing values in a feature that jeopardized the integrity of the data was removed. Categorical features such as financial information that contained skewed data were consolidated through logarithmic transformation. Other features such as Race which contained small subgroups that contained lower numbers were consolidated to 'Other' and not removed to ensure data usability. New features were drawn out from existing features.

For example, international status was extracted from home addresses and multiple features were extracted from financial aid information. Final dataset consisted of 2464 students records with 2115 retained students whereas 349 ceased students.

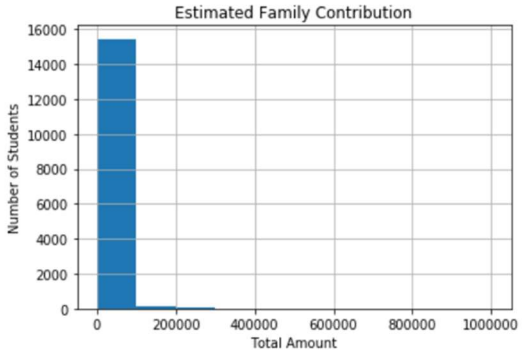


Figure 4: Heavily skewed data with large number of missing values for Estimated Family Contribution (EFC).

4.2 Data Preprocessing and Exploration

Categorical features that were ordinal such as Class was label encoded whereas non-ordinal, dummy variables were used. This was done cautiously as label encoding creates an imposed ordinality when applied to features not suitable for it. In this project such examples were Ethnicity, Race, Campus location. However, Decision trees and Random Forest algorithms are able to handle label encoded variables regardless of the ordinality issue. Simple plots were created to analyze the data. After preprocessing, there were 58

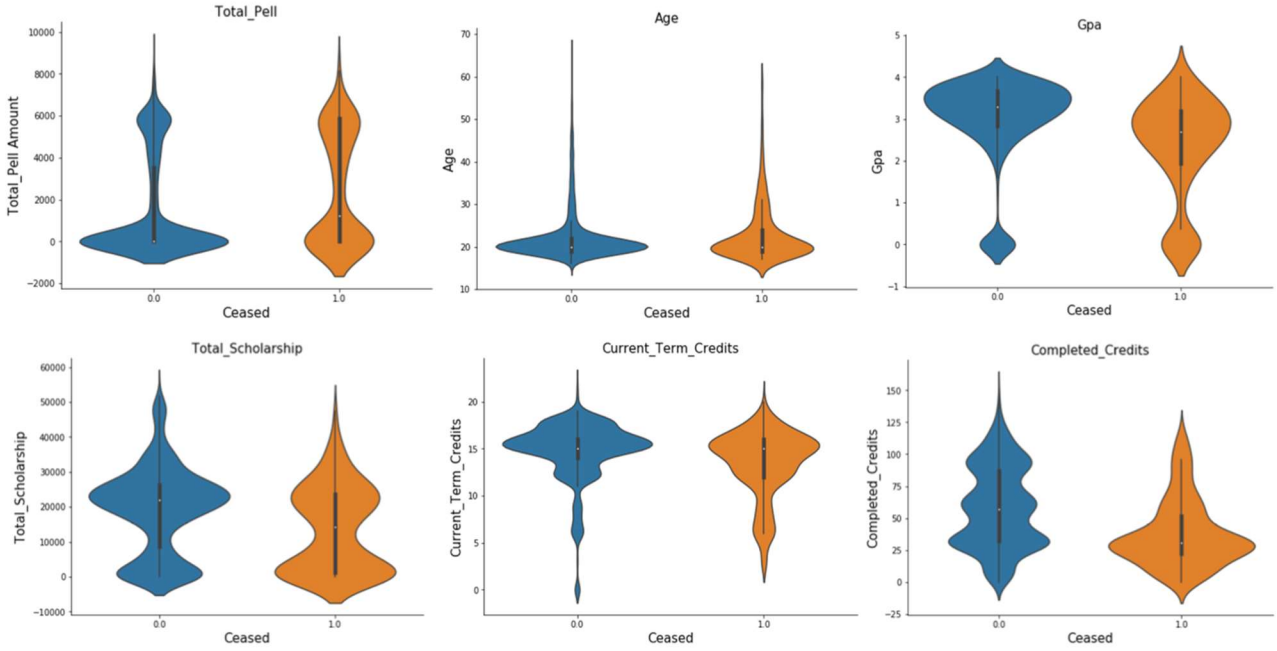


Figure 5: Violin plots univariate analysis of quantitative features showing comparison between ceased vs retained students.

number of missing values. Due to the same reason, imputation was not done.

Exploratory analysis with the features was done which concluded that there was clear relationship between the features and retention. For example: Univariate plots were

done to observe any relationship. These plots are based on a single variable and show the frequency of unique values of

a given variable. In a univariate plot, the y-axis is the probability density function. In this method, a continuous curve (the kernel) is drawn at every individual data point and all of these curves are then added together to make a single smooth density estimation. This process was fundamental towards analyzing outliers and skewed data. Violin plot was done to visually represent the result. A violin plot is a method of plotting numeric data. It is similar to box plot, with addition of a rotated kernel density plot on each side. Features such as credit load and total Pell amount showed an apparent difference in observation between the classes.

Table 1: 25 features that were extracted for modeling.

5. METHODOLOGY

26. The distribution of the data between the classes was significantly imbalanced: 2115 retained students and 349 ceased students with the ratio of 85% to 10%. In imbalanced classes, one can get high overall accuracy, but it does not necessarily generate a sound model. The overall accuracy might be high, but for the minority class (here ceased students), there can be a very low recall score (sensitivity). As we want to have a high recall i.e. correctly identify students who are at risk of ceasing enrollment, it is necessary to have a balanced dataset. In order to evaluate the change in model performance with regards to the number of sampling for classes, three different approaches were taken:

a) The data was left unbalanced. This was done intentionally to compare the performance with other models.
b) Random under sampling was done for retained students. It was done by randomly picking samples with or without replacement. The retained students that contained of 2115 records were under sampled to 349.
c) Over sampling was done for ceased students (minority class) through SMOTE. The 349 records of ceased students were increased to 2115 through SMOTE.

Features	Data Type	Description
Class	Multi Nominal	School year of the student
Completed_Credits	Numerical	Total completed credits
Gpa	Numerical	Cumulative grade point average
Ethnicity	Multi Nominal	Ethnicity of the student
Race	Multi Nominal	Race of the student
Credits_Load	Multi Nominal	Credit load categorization
Age	Numerical	Age of the student
Admit_Status	Multi Nominal	Status of the student during admission
Current_Term_Credits	Numerical	Total Credits in the current term
Total_Scholarship	Numerical	Total amount of awarded Scholarship
Total_Pell	Numerical	Total amount of awarded Pell Grant
Pell_Eligibility	Binary Nominal	Pell eligibility
Loan	Binary Nominal	Loan taken or not
Work	Binary Nominal	Work study student or not
Major	Multi Nominal	Major of the student
Campus_Location	Multi Nominal	Campus location of the student
Gender	Binary Nominal	Sex of the student
Current_Student_Type	Multi Nominal	Type of Student status
International	Binary Nominal	International or not
In_State	Binary Nominal	In State or Out of State
Residence_Type	Multi Nominal	Housing Location(on, off, commute)
Ceased	Binary Nominal	Class variable (0,1)

5.1 Model Selection

For the three different approaches, classifiers: Logistic Regression, Random Forest Classifier, Decision Tree Classifier and Support Vector Classifier were chosen. For evaluation, accuracy, precision, recall, ROC and F1 scores were considered. It is important for the model to have a high recall (sensitivity) as they should efficiently predict ceased students. Sensitivity is the ration of True Positive (TP) by the sum of True Positive (TP) and False Negative (FN) which is shown by the equation 2.

$$Sensitivity(Recall) = \frac{TP}{TP+FN} \quad (2)$$

6. RESULTS

Classifier models were evaluated based on various metrics. Below are the tables that evaluate the models.

6.1 Tables

The values for evaluation metrics are below:

Model	Accuracy	Precision	Recall	ROC AUC	F1
Decision Tree	0.66	0.61	0.71	0.66	0.71
Random Forest	0.78	0.73	0.82	0.78	0.77
Logistic Reg	0.71	0.72	0.65	0.72	0.68
SVM	0.63	0.61	0.63	0.63	0.62

Table 2: Evaluation metrics for Under Sampled of retained students (Ceased 0)

Model	Accuracy	Precision	Recall	ROC AUC	F1
Decision Tree	0.84	0.82	0.85	0.84	0.83
Random Forest	0.91	0.93	0.89	0.91	0.91
Logistic Reg	0.74	0.75	0.69	0.74	0.72
SVM	0.63	0.61	0.63	0.63	0.62

Table 3: Evaluation metrics for Over Sampling/SMOTE of ceased students (Ceased 1)

Model	Accuracy	Precision	Recall	ROC AUC	F1
Decision Tree	0.86	0.5	0.04	0.51	0.08
Random Forest	0.79	0.27	0.30	0.58	0.28
Logistic Reg	0.86	0.0	0.0	0.49	0.0
SVM	0.86	0.0	0.0	0.50	0.0

Table 4: Evaluation metrics for Imbalanced data (Ceased 349, Retained 2115)

6.2 Explanation

The performance of the model with imbalanced data was the poorest. The accuracy scores for imbalanced data were 0.86, 0.79, 0.86 and 0.86 for Decision Tree, Random Forest, Logistic Regression and Support Vector Classifiers respectively. However, the result from other evaluation metrics were poor. This is due to the imbalanced nature of the data that caused the algorithm to have a bias. For the purpose of this study, it is important to consider high recall (sensitivity) score. Hence it is imperative to refer to other metrics rather than just accuracy for proper evaluation. The above observation is reflected in Figure 6.

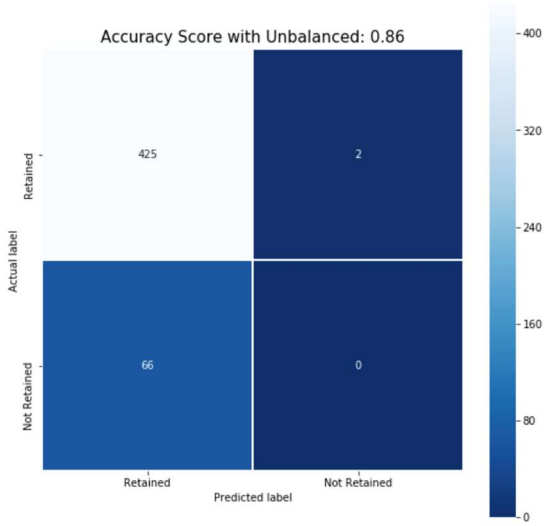


Figure 6: Logistic Regression Confusion Matrix for Unbalanced Data

Both Under sampling and SMOTE Oversampling techniques performed significantly better with recall scores of .82 and .89 respectively.

The best model from the above methods was over sampling with SMOTE of ceased students. Specifically, Decision Tree with SMOTE performed the best. For this study, it is important to assess various performance metrics in order to conclude with a final model for deployment. Hyperparameter tuning was done with random search to generate the best model.

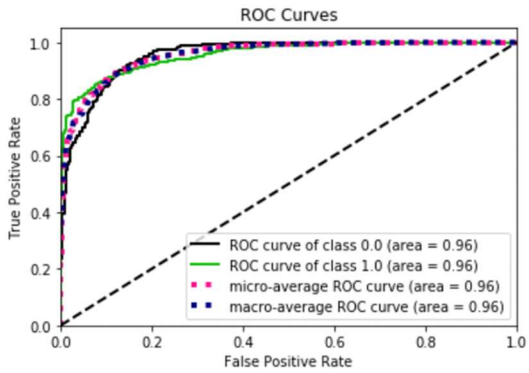


Figure 7: ROC curve for Decision Tree with SMOTE

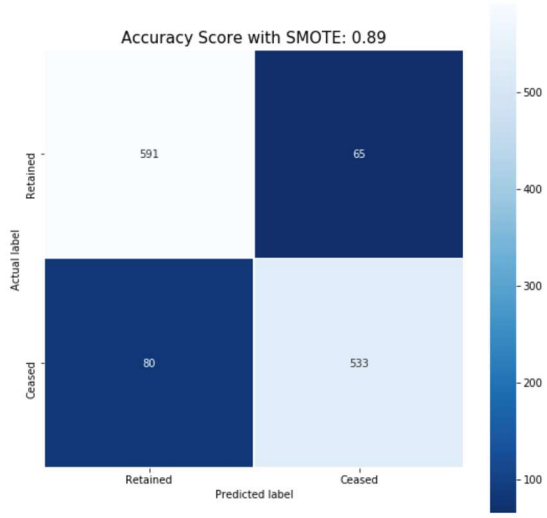


Figure 8: Confusion Matrix for Decision Tree with Smote.

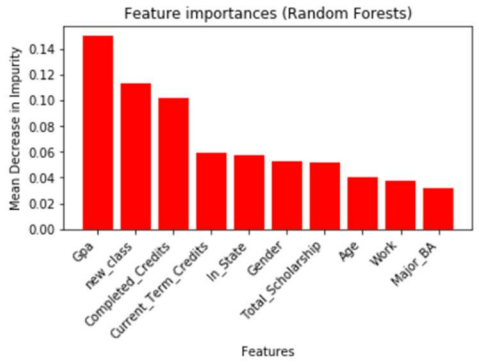
6.3 Feature Importance

Important features for the modeling of Random Forest with SMOTE were extracted. These features can be utilized to build intervention models. Gini Importance or Mean Decrease in Impurity (MDI) calculates each feature importance as the sum over the number of splits (across all tress) that include the feature, proportionally to the number of samples it splits. Feature importance from Scikit-learn library does not account for test data. It shows features that were considered the most important while training (exclusive of any relation with test data). To account for features affecting the test data, permutation importance was evaluated. Permutation importance is calculated after a model has been fitted. It answers the following question: If we randomly shuffle a single column of the test data, leaving the target and all other columns in place, the columns with the highest level of variability would be of the highest importance.

For the purposes of this study, permutation with recall change was done with 100 iterations to get the most important features.

Figure 9: Top 10 features from Random Forest (SMOTE)

After applying both methods, class level, GPA, completed credits, gender, total scholarship, in-state residence, total Pell grant were



the most salient features.

Feature	Recall Percentage
new_class	5.188679
Gpa	4.169811
Gender	1.396226
Total_Scholarship	1.396226
In_State	1.226415
Current_Term_Credits	1.188679
Total_Pell	0.981132
Completed_Credits	0.698113
Residence_OnCampus	0.584906
Loan	0.433962

Table 5: Permutation features with Recall percentage

7. APPLICATION AND DISCUSSION

The probability obtained using random forest is not a true probability estimate. It reflects the proportion of votes of the trees in the ensemble. Logistic regression is an extremely efficient mechanism for calculating probabilities. In context of this project, the logistic regression of the best performing model: SMOTE can be utilized to create a probability of a student ceasing enrollment. This is done by calculating the probability on the test data which is unseen by the algorithm. Therefore, for upcoming students or current students, the model can be applied to identify 'at-risk' individuals to take necessary measures. Important features can be studied to create intervention protocols to ensure student churn. This probabilistic measure was applied to unseen test data to identify 'at-risk' students.

	True Ceases	Predicted Ceases	Cease_risk
3009	1.0	1.0	0.97
4102	1.0	1.0	0.97
3700	1.0	1.0	0.96
2736	1.0	1.0	0.96
3656	1.0	1.0	0.95
552	0.0	1.0	0.95
3007	1.0	1.0	0.95
3234	1.0	1.0	0.95
4206	1.0	1.0	0.95
4008	1.0	1.0	0.94
631	0.0	1.0	0.94
2507	1.0	1.0	0.93
3543	1.0	1.0	0.93
2533	1.0	1.0	0.93
3889	1.0	1.0	0.93
2679	1.0	1.0	0.93
3629	1.0	1.0	0.93
4125	1.0	1.0	0.92
2676	1.0	1.0	0.92
3611	1.0	1.0	0.92

Tables 6: Ranking students who are at risk of ceasing enrollment. True cases were compared with predicted cases. High level of accuracy was observed based on the risk scores.

8. CONCLUSION

The project highlighted preacautionary measures to tackle imbalanced dataset. Initial accuracy scores were erroneous and was not considered for deployment bfore consdiering other evaluation metrics. For the purpose of this study, recall score/sensitivity was more salient than accuracy. Both, undersampling and oversampling were very effective in tackling the problem compared to the imbalanced data. The SMOTE model was most effective in predicting true ceases while performed on the test data of 2017. The ultimate practical goal of this project was able to identify 'at-risk' students which it was succesful in doing. This is important as the model can be deployed to identify at risk students for the future classes. This model can be applied to encourage intervention efforts to mitigate student ceases.

9. LIMITATIONS AND FUTURE WORK

Due to the inconsistencies of data, the model has not been tested on other years. The test data consisted of data from the same year (2017). To get a better performance evaluation, it is necessary to test the model in newer true data instead of the test data of the model. To do so, further data collection (cleaning and features engineering) should be done. Other features concerning students such as club affiliations, disciplinary records, in-campus involvement, etc can be added to the model to bolster the scope of effictiveness. After succesful perfomance evaluation on newer data, the model can be deployed for practical usage.

10. REFERENCES

[1] Graduation and Retention Rates - What is the full-time retention rate in postsecondary institutions? https://nces.ed.gov/ipeds/TrendGenerator/app/answer/7/32. Accessed: 2020-04-17.

[2] Lau, L.K. 2003. Institutional Factors Affecting Student Retention. Education. 124, 1 (Sep. 2003), 126.

[3] Seidman, A. 2012. College Student Retention: Formula for Student Success. Rowman & Littlefield Publishers.

[4] Tinto, V. 2006. Research and Practice of Student Retention:What Next? Journal of College Student Retention: Research, Theory & Practice. 8, 1 (May 2006), 1–19. DOI:https://doi.org/10.2190/4YNU-4TMB-22DJ-AN4W.

[5] Alkhasawneh, R. Developing a Hybrid Model to Predict Student First Year Retention and Academic Success in STEM Disciplines Using Neural Networks. 135.

[6] Whitlock, J.L. Using Data Science and Predictive Analytics to Understand 4-Year University Student Churn. 187.

[7] Murtaugh, P.A., Burns, L.D. and Schuster, J. PREDICTING THE RETENTION OF UNIVERSITY STUDENTS. 17.

[8] Rajuladevi, A. A Machine Learning Approach to Predict FirstYear Student Retention Rates at University Of Nevada, Las Vegas. 70.

[9] Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P. 2002. SMOTE: Synthetic Minority Over-sampling Technique. J. Artif. Intell. Res. (2002). DOI:https://doi.org/10.1613/jair.953.

[10] He, H. and Ma, Y. 2013. Imbalanced Learning: Foundations, Algorithms, and Applications. John Wiley & Sons.

About the author:

Udip Bohara is a graduate student at Mercyhurst University.

Location, Location, Location: A Linguistic Analysis of Tweets Using Machine Learning and Natural Language Processing Techniques

Daniel J. Bahntge
Department of Computing & Information Science
Mercyhurst University
Erie, PA, USA
dbahnt92@lakers.mercyhurst.edu

ABSTRACT

This paper presents an analysis of social media textual data using natural language processing (NLP) and machine learning (ML) techniques. Commercial NLP applications rely on ML models trained with many features to make their predictions, and this requires a lot of engineering effort to define, compute, and select appropriate features. Much of this challenge stems from text classification. The task of pulling textual data from multiple sources across the world would require a substantial amount of time and resources. This study builds on previous linguistic research that uses georeferenced language mapping techniques on social media English textual data collected from the microblogging website Twitter. This study: i) performs stylometric analysis on 3 samples of tweets collected outside the contiguous United States; ii) tests 4 ML models through the lens of a multi-class classification problem to determine which model best predicts an English variety using a held-out collection of tweets; and iii) tests 4 ML models through the lens of a binary classification problem to gain insight into the inner workings of each model, and measures the contributions of their linguistic characteristics.

Keywords

Natural language processing (NLP), machine learning (ML), data science, social media mining, social media text analysis, stylometry.

27. INTRODUCTION

NLP is a collective term that refers to the automatic computational processing of human languages. NLP is defined as the field of designing methods and algorithms that take as input or produce as output unstructured, natural language data [38].

As businesses continue to see an influx of data, it is important for them to invest in research that incorporates predictive capabilities. Businesses are constantly presented evolving challenges due to dynamic environments and complex decision-making processes. With evolving business processes and evolving technologies, businesses and consumers will face evolving forms of communication.

Every business is in the communication business. The most successful businesses are those that can view the business transactions through the eyes of the consumer and can effectively communicate their message to their consumers. The experience and message for the consumer needs to be accurate and relevant. The context of the message being communicated ought to be the focus. It is crucial that businesses allocate their marketing and advertisement resources to data-driven research and development to better hear what their consumers are saying. One of the most effective ways to achieve this is to analyze the language consumers are using. Language allows for the transmission of ideas, thoughts, and emotions from one person, or group, to another. Language is fundamental to what it means to be human, yet language is constantly changing as new words enter and leave our collective dictionaries. Social media microblogging platforms have produced a variety of ways in which language has been used for communication.

The collection of social media textual data has started a new kind of conversation among consumers and companies, creating new opportunities for organizations to understand their consumers. Social media micro-blogging services, such as Twitter, offer an immense resource of real-world linguistic data. Tweets are publicly available and thus provide an enormous resource of authentic linguistic text from people all around the world. First, we will examine the history of NLP approaches that have utilized large corpora of English textual data. Second, we present current applications (from a corpora-based perspective) within the fields of NLP and ML. Third, building off characteristics of previous research [21] that focuses on aspects from dialectometry, we create a smaller-scale representation of linguistic regional variation using Twitter textual data (using the Twitter API) to show how its methods can be adapted to more traditional problems of dividing countries into linguistic regions.

The first part of this paper incorporates techniques derived from stylometry, using the z-score statistic and the John Burrows Delta statistic. The delta statistic was chosen to measure the similarity of writing style on 3 samples of Tweets located outside the contiguous United States to determine which linguistic region of the United States they are most like. These 3 samples of tweets were collected from Honolulu, Hawaii, San Juan, Puerto Rico, and Anchorage, Alaska. Findings from this stylometric analysis are discussed in the Results section (3.4).

Part II of this research will divide the United States into 4 linguistic regions, and train 4 ML models through the lens of a multi-class classification problem to predict a linguistic region using a held-out collection of tweets. Using the same 4 regions in Part II, Part III of this research will test 4 ML models through the lens of a binary classification problem to gain insight into the inner workings of our models to measure the contributions of the linguistic characteristics proposed in this paper.

This paper ultimately provides groundwork for future research of social media linguistic analysis, with hopes it can inform future research on the variation that occurs within regions of English-speaking countries, as well as the linguistic characteristics found on microblogging websites such as Twitter.

28. RELATED WORK

28.1 Data Science and Textual Analysis

Textual categorization has been described as the intersection of ML and information retrieval [38]. There has been considerable research using complete global representations of linguistic regional variation as a form of textual classification [20, 21, 24, 25, 33, 39, 41, 42]. An important question raised when classifying text is what features should be considered to measure regional differences in linguistic structure.

Within NLP literature, the problem of regional difference is typically approached through dialectal variation and performing dialect identification (DI) to measure lexical diversity within a text. Some studies [30, 31, 32] have used phonological features to determine varieties across languages. Others [33, 34, 35] have studied language-dependent variations that have ranged from a few features to hundreds of features. Yet other studies [17, 18, 36, 37] have considered language-independent representations, such as function words, and part-of-speech labels.

The consensus in dialect identification (DI) is there are lexical and syntactical characteristics that have been proven to be consistent within languages across multiple data sources [1]. Much of this research classifies linguistic regional variation using multiple languages and multiple sources, however this research will focus on the English language and a single source, Twitter textual data, and will combine techniques (linguistic characteristics) found in previous research.

28.1.1 Stylometric Analysis

Exploring linguistic characteristics through the style in which text is written is an area of research that has been studied extensively [51, 52, 53, 54]. Stylometry is the study of literary style through computational reading methods and is based on the underlying premise that an individual writers' style is relatively consistent through time. These techniques have been used to study the difference between how men and women write, detect plagiarism, and, most notably, for authorship attribution. Stylometric techniques will be implemented in this research to present a novel way of measuring linguistic characteristics and linguistic variation of English throughout the United States on the microblogging website Twitter (3.1.1).

Understanding the relationship between words of the language is important in addressing the problem of textual inconsistencies found in this dataset. The statistical measures chosen for Part I, will allow us to: i) measure the overall variation of our presented semantic characteristics within the contiguous United States; and ii) compare the variation of semantic characteristics of a sample of English tweets outside the contiguous United States.

Having a good semantic understanding will be important because: i) it can help address the highly ambiguous, variable, evolving nature of language; ii) it will allow us to consider suitable statistical metrics to evaluate our linguistic regional varieties; and iii) it will allow us to address the textual inconsistencies found in our models.

28.1.2 Language Modeling: Interactive Effects and Important Interactions

There are many considerations to keep in mind when conducting any textual analysis. It is incredibly challenging in the field of computational linguistics to explain linguistic variation in historical or social terms [21]. For example, what real-world events caused the spread of English to create the linguistic regional variation? Not only are there limitations to what is knowable at any given point in time, but also limitations to what can be inferred from the language of the past (e.g. background knowledge of language changes, semantic drift, etc.), as well as language cohesion on a global level.

The English language is more approachable than other languages because it is not as morphologically sophisticated (i.e. simpler word structures) as other languages and offers more computational resources across the world (conversely, offering far less academic research). Focusing only on English within the United States will allow a smooth integration of spatial information in our ML model evaluation.

Since the classes of this research are pre-defined (consisting of tweets with known georeferenced locations) and involve textual data from a single source (Twitter), we can explore the semantic space (semantic characteristics) of our language more effectively due to there being less potential spatial inconsistencies within the data. Focusing only on Twitter data will avoid potential inconsistencies that can arise from the aggregation of information from multiple data sources [13, 14, 15, 16].

It should be noted, this paper is not a deep dive into dialectometry, which primarily pays attention to spoken speech [21]. Though phonological aspects of our text are not explored here, the same features used in dialectometry can be implemented to define written dialectal features of a language. Therefore, the distinction between dialects and varieties is not made. One could just as easily have named the linguistic regions: regional dialects (or: linguistic regions).

The approach presented in this paper is to answer the question of whether the distinction of one linguistic regional variety to another can be derived using data-driven language mapping techniques on a small scale. A second distinction is occasionally made between dialects and varieties; the conceptual division of inner-circle, outer circle and expanding circle dialects [21]. For the sake of this paper, both dialect and variety distinctions will be approached empirically (i.e. easily interpretable regional divisions: MW, NE, S, W). A third distinction is often made in linguistic analysis, and that is between places (English used in the United States) and varieties (American English). It has been found that in corpus-based research, this assumption is problematic [21]. A farmer born and raised in Erie is assumed to be a local, a representative of American English; an IT expert born in Cambodia but educated and living in Erie is not. In the context of social media textual analysis, this distinction might be made between user specified location metadata and actual georeferenced location metadata. This paper makes no effort to exclude participants within a given geo-referenced location; American English is simply English used in the United States.

To clarify all 3 distinctions typically associated with linguistic analysis, a linguistic regional variety will henceforth be referred to as an English Variety.

Although these three distinctions are typically made and referenced when conducted on a global scale, it is important to consider linguistic interactions from a global perspective. More relevant to

this research, [11, 12] have shown the aggregation of tweets can predict geo-location in grid-based representations of the United States. Text within social media communication is often more vernacular [3] and is more likely to reveal the influence of geographic factors than text written in a more formal medium of communication, such as news text [9]. Thus, this paper aims to predict regional differences using pre-determined regions (MW, NE, S, W) using textual data with known geo-referenced meta-data (though not explicitly encoded as model features).

Not only will highlighting these interactions allow us to better frame the linguistic characteristics considered within this paper and let us know the desirable elements to consider in the context of this experiment but will allow future researchers to consider the many interactions that exist within NLP research more broadly.

28.1.3 Language Modeling

28.1.3.1 Binary versus Multi-Class Classification

To predict an English Variety, we will implement 4 ML classification algorithms to measure precision, recall, F1-score and accuracy. Part II of this research will explore the English Varieties from a multi-class perspective to predict a linguistic region and measure model accuracy. Part III of this research will explore the English Varieties through the lens of a binary classification problem to determine which algorithm is best at uncovering the linguistic characteristics of our textual data.

I leave it to future research to test different classification methods and combine other linguistic and background knowledge resources to identify other linguistic relationships of the textual data presented in this paper. It is important to note, engineering an NLP pipeline that can effectively address the evolution of language at a granular level, will help address the myriad of possible downstream NLP tasks that may exist.

28.1.3.1.1 Support Vector Machines (SVMs)

Support Vector Machines (SVMs) achieve predictions by maximizing margins between classifications. This means that training examples are transformed onto a hyperplane that increases the distance from one class to another. Optimally, the training examples that are closest to the maximum margin are called the support vectors. SVMs are generally accepted as the standard within NLP classification research and literature [22, 23, 29, 38], therefore will be the first algorithm used to predict an English Variety.

For each variety, the SVM will use the training data to learn the weights for each construction of our linguistic characteristics. The model builds a high-dimensional representation of each variety that maximizes the distance between them. This is accepted as the most efficient approach in classification problems where very high-dimensional spaces are the norm.

Part II of this research will implement scikit-learn's Support Vector Classifier with kernel type: 'rbf'. Part III will implement scikit-learn's Support Vector Classifier with kernel type: 'linear' [49]. The quality of these models will be evaluated on held-out testing data.

28.1.3.1.2 Decision Tree Classifier

The second algorithm we will be testing is the Decision Tree Classifier using scikit-learn's DecisionTreeClassifier [4]. Decision trees are a non-parametric supervised learning method used for both classification and regression. The classifier creates a model that predicts a target variable using decision rules inferred from the input features. The advantage of using this classifier in the context of this research, is that it offers easily interpretable rules about the decisions it makes when classifying our linguistic characteristics.

28.1.3.1.3 Random Forest Classifier

The third algorithm that we will be testing is the Random Forest Classifier using scikit-learn's RandomForestClassifier [2]. A random forest classifier is another supervised learning approach that fits many decision tree classifiers on various sub-samples of our dataset and uses averaging to improve the predictive accuracy as well as control over-fitting. The advantage of using this classifier in the context of this research, is that it allows for exploratory analysis on the performance measures of our models. This is a big advantage when measuring the degree to which our linguistic characteristics contribute to performance metrics.

28.1.3.1.4 Extra Trees Classifier

The fourth algorithm that will be used will be the Extra-Trees Classifier using scikit-learn's ExtraTreesClassifier [26]. This classifier is another supervised learning approach that fits several randomized decision trees (extra-trees) on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

29. METHODS

29.1 Linguistic Similarity

29.1.1 3 Stylometric Signatures

Incorporating stylometric techniques is the first part of our linguistic analysis. The purpose of implementing stylometric techniques is: i) it is the most prominent method used within stylometric analysis; and ii) stylometric techniques offer a novel analysis of the tweets that are located outside of our 4 linguistic regions explored in Part II and Part III of this paper. Stylometric analysis will allow us to test a sample of tweets queried from outside the contiguous United States and show us the degree to which the semantic characteristic features selected for our ML models represent the regions used later in our study.

The John Burrows Delta statistic, *see Figure 1*, is typically used to measure the distance between a text whose authorship is being compared to another corpora of text. Though this is a technique often found in authorship attribution, this statistic is well suited for the purposes of this paper and will be utilized to ascertain a sample of tweets collected from 3 geographical locations (Honolulu, Hawaii, Anchorage, Alaska and San Juan, Puerto Rico) outside the contiguous United States that comprise our English varieties.

$$\Delta_c = \sum_i \frac{\left| Z_{c(i)} - Z_{t(i)} \right|}{n}$$

Figure 1: Equation for John Burrows' Delta statistic

The Delta statistic will measure how the 3 geographical locations diverge from the average of all 4 regions. Features will be given equal weight, which will avoid the problem of common features having influence over results. To calculate the stylometric Delta scores, a six-step process is used:

i. **Feature Selection:** A total of 50 features were used as our "standard". These features comprised of the 50 most common char-grams found in all 12,000 tweets that comprise our 4 linguistic regions.

ii. **Contribution of each Feature:** Calculate the share of each regions' tweets that are represented by the features selected as a percentage of the total number of features present in all tweets.

iii. **Feature Averages and Standard Deviations:** Calculate the mean and standard deviations of the frequencies expressed in each region. (The contribtribution of a regions' corpora has equal representation.)

iv. **Sub corpora z-scores:** For each feature of each region the z-score, *see Figure 2*, is calculated to measure the distance from all 12,000 tweets each feature in each region happens to be.

v. **3 Test Case z-scores:** Calculate the same z-score for each feature in each of our 3 sample geolocations.

vi. **Delta Scores:** Finally, we calculate the Delta scores of our 3 sample geolocations. The lowest Delta score tells us which English Variety our 3 samples most likely belong to.

$$Z_i = \frac{C_i - \mu_i}{\sigma_i}$$

Figure 2: Equation for the z-score statistic

29.2 Language Modeling

29.2.1 Exploratory Analysis of Data Sources

Next, we incorporate data from a single source of linguistic data, Twitter, which gives text from archived tweets. The advantage of using a microblogging website like Twitter is the accessibility of geo-referenced metadata utilized to designate each English Variety. Studies, [19, 44] have shown there is a relationship between geo-referenced textual data and linguistic variation. To determine an English Variety, Python was used to wrangle tweets from Twitter (using the Twitter API). A spatial search was used to collect Tweets between January 27, 2020 and February 6, 2020. From there, 47 cities with a population of at least 50,000 were selected at random within the contiguous United States, *see Figure 3*. The coordinates for each of the 47 cities were queried for any tweet within a radius of 25km.

Nearly 5,000,000 tweets were queried for this research. An arbitrarily high number (100,000) of tweets was assigned when querying from each geographical location. Each tweet queried was collected from a separate user, as to avoid the possibility of querying a large number of tweets from a single account (thus, avoiding bot accounts), so that no single user had an influence over the language being used in a given region.

Of the nearly 5,000,000 tweets collected, only 0.0027 of those tweets included geo-referenced meta-data (excluding 'bounding-box' referenced tweets). This city-based search avoids biasing the selection (e.g. using region-specific keywords or hashtags). While this approach avoids a regional bias, it could underrepresent rural areas given the 25km radius of each collection, *see Figure 4*.

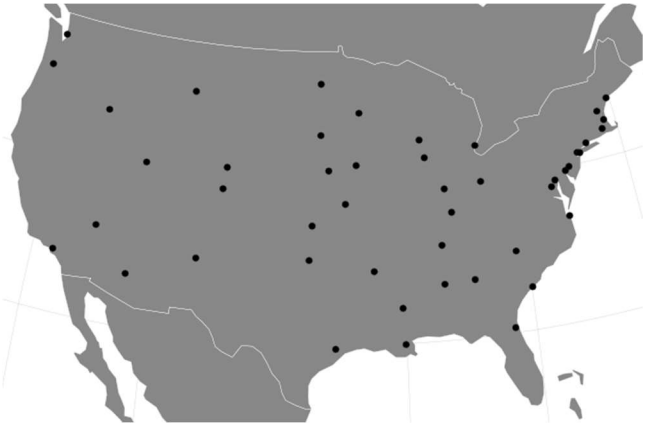


Figure 3: All 47 US Cities used for Language Modeling

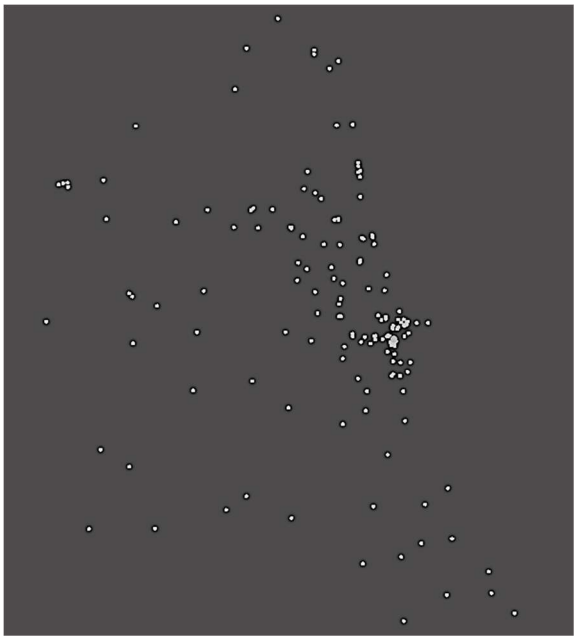


Figure 4: All Tweets Collected within 25km of Chicago

```
[('NN', 61954),
 ('JJ', 19976),
 ('VBG', 4339),
 ('VBP', 4134),
 ('RB', 4002),
 ('IN', 3107),
 ('VB', 3011),
 ('VBD', 2297),
 ('JJS', 1446),
 ('NNS', 1194)]
```

Figure 5: Top 10 Most Frequent POS-Tags

A total of nearly 3,500 tweets were queried per region. Then 3,000 were selected at random from each region. Taking this approach: i) ensures each class is balanced for Part II evaluation; and ii) gives the most accurate assessment of which linguistic characteristics contribute to our performance metrics.

The tweets were queried and stored in JSON formatted documents. The tweets were then preprocessed using the NLTK toolkit. Each tweet was assigned a sentiment score, using the Vader SentimentIntensityAnalyzer compound score [10]. Each tweet was lowercased and tokenized using the NLTK TweetTokenizer. Hashtags, URLs, RTs (retweets), and @ symbols were then removed. The tweets were filtered to include only those that included the English language. Each tweet was treated as a sentence and tagged using the NLTK POS-Tagger. The bigrams, trigrams, fourgrams, 3-char grams and 4-char grams were also calculated for each tweet.

The 50 most frequent features from each region, the 50 most frequent features from all tweets (semantic characteristics), the top 10 most frequent POS-Tags (syntactic characteristics, *see Figure 5*) and the Vader score (sentiment characteristic) were the features that populated the 4 ML models. The hope of this research is to find insight into the language used within the United States and analyze the interactions of these linguistic characteristics, *see Figure 6*.

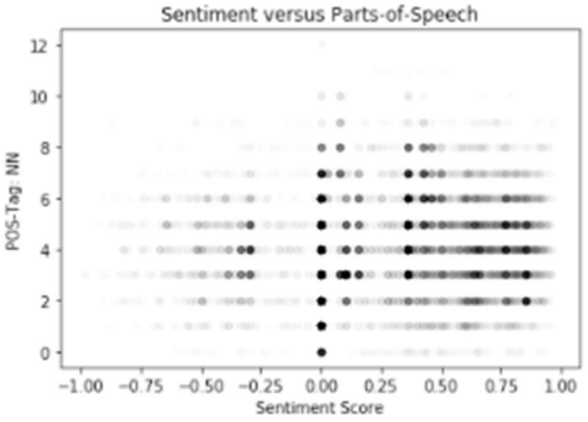


Figure 6: Sentiment versus Parts-of-Speech

29.3 English Variety Classification

This research takes a supervised learning classification approach to determine an English Variety. There are a couple of advantages in taking this approach: i) the model can be evaluated using prediction accuracies on held out testing data; ii) a classification approach offers an implicit measure of the degree of linguistic variation (e.g. uniqueness of syntactic and semantic construction) between varieties.

The 4 English Varieties for this research (*see Figure 7*):

i. **Mid West (MW):** Ohio, Michigan, Indiana, Illinois, Missouri, Iowa, Wisconsin, Minnesota, Kansas, Nebraska, South Dakota, North Dakota

ii. **North East (NE):** Pennsylvania, New York, Maine, Vermont, New Hampshire, Massachusetts, Connecticut, Rhode Island

iii. **South (S):** West Virginia, Virginia, Delaware, Maryland, Kentucky, North Carolina, South Carolina, Tennessee, Georgia, Alabama, Mississippi, Florida, Louisiana, Texas, Arkansas, Oklahoma

iv. **West (W):** Washington, Oregon, Idaho, Montana, Wyoming, California, Nevada, Utah, Colorado, Arizona, New Mexico

The 12,000 tweets were split into training and testing (80/20). The training data was used to train a support vector classifier, a decision tree classifier, an extra-trees classifier and a random forest classifier. The resulting models were then used on the remaining testing data.

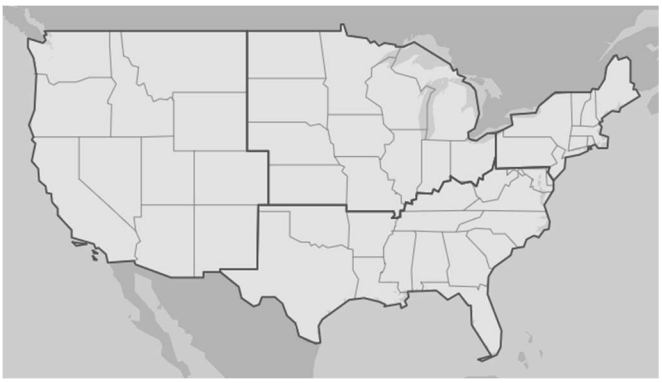


Figure 7: The 4 English Varieties

29.4 Results

29.4.1 Part I: Stylometric Signature Performance

This research found that stylometric techniques have promising implications for social media textual analysis and social media mining that incorporate georeferenced metadata. Honolulu, Hawaii, and San Juan, Puerto Rico were most like the South region. The most surprising finding of this part of our research were the delta scores of Anchorage, Alaska, and Honolulu, Hawaii that were most like the South region. One would suspect that Anchorage would be most similar to the West region (nearest geographical location), but in fact had semantic characteristics more similar to the Mid West and North East regions than it did the West region, *see Figure 8*.

		delta score
	MW	14.089
(Anchorage,	NE	14.215
Alaska)	S	13.673
	W	14.309
	MW	14.233
(Honolulu,	NE	14.202
Hawaii)	S	13.651
	W	14.555
	MW	13.997
(San Juan,	NE	14.007
Puerto Rico)	S	13.64
	W	13.995

Figure 8: Stylometric Delta Scores

29.4.2 Part II: Multi-Class Classification Performance

The goal of this section is to assess the prediction accuracy across English Varieties and the similarity between each English Variety. We can best interpret the accuracy of our models by considering the global behavior of each model across all target classes, *see Figure 10*. The accuracy scores ranged between 0.38 and 0.413, with the random forest classifier being the most accurate of the 4 models.

The goal of this research was not to achieve best accuracy but to peel back the underlying linguistic characteristics and the degree to which our chosen linguistic features contribute to other performance metrics to uncover the inner workings of our models and their performance on each linguistic characteristic.

All 4 models recorded poor accuracy scores on our validation sets/ cross-validation scores. Fine-tuning the hyperparameters of our models resulted in similar accuracy scores with insignificant changes in precision, recall, and F1-scores. These results indicate that there is not enough linguistic variation within our regions for our models to capture.

Therefore, the next step in our research was to turn our multi-class classification evaluation into a binary class classification. Though accuracy of our models were low, treating each region as a binary classification problem can offer valuable insight into the models best used for each linguistic characteristic.

```
[[1445  331]     [[1766   10]
 [ 414  210]]     [ 611   13]]
```

(a) (b)

Figure 9: MW Confusion Matrix of Decision Tree Classifier

		precision	recall	f1-score	accuracy
	MW	0.36	0.44	0.39	0.38
(SVM)	NE	0.43	0.42	0.43	
	S	0.34	0.29	0.31	
	W	0.42	0.38	0.4	
	MW	0.38	0.38	0.38	0.395
(Decision-	NE	0.43	0.43	0.43	
Tree)	S	0.38	0.33	0.36	
	W	0.39	0.43	0.41	
	MW	0.39	0.42	0.4	0.406
(Extra-	NE	0.47	0.43	0.45	
Trees)	S	0.36	0.34	0.35	
	W	0.41	0.44	0.42	
	MW	0.39	0.4	0.39	0.413
(Random-	NE	0.49	0.45	0.47	
Forest)	S	0.36	0.37	0.37	
	W	0.42	0.44	0.43	

Figure 10: Multi-Class Classification Performance Scores

29.4.3 Part III: Binary Classification Performance

Next, 4 models were then trained, this time as a binary classification problem. After training and fine-tuning our models, the model that recorded a significant improvement in accuracy was the decision tree classifier fitted on the West region. When tuned to a max_depth of 4, criterion of entropy and number of components set to 19, the model had a 5% increase in accuracy, but recall dropped by 33%, showing poor retrieval for the relevant target class (W).

This hyperparamter tuning highlights the precision-recall tradeoff and shows accuracy may not be a relevant performance metric of social media textual analysis in all contexts. Though the accuracy proved highest from a per-class evaluation (*see Figure 12*), if we examine the confusion matrix of a decision tree classifier (*see Figure 9*) and its performance on the Mid West region; had we maximized the accuracy via hyper-parameter tuning (*Figure 9.b*), we would have missed valuable insights of our ML models (*Figure 9.a*).

The main consideration now being imbalanced classes, we will next examine the F1-score of each individual English Variety, *see Figure 11*. This will allow us to evaluate our models and their internal properties. The higher the scores for each English Variety, the more distinct the variety (i.e. linguistic characteristic structure). This allows us to evaluate which regions have the most similar linguistic characteristic profiles. The higher the value, the harder the model is trying to distinguish between an English Variety (i.e. varieties more like each other). One possible reason could be that these varieties have greater influence amongst each other. This shows that each model not only distinguishes between English varieties but can also situate the varieties in relationship to one another. More varieties (i.e. greater lexical diversity) may be present if textual data was collected from other social media microblogging websites, or had we split the 4 regions into smaller geographical locations.

	(SVM)	(Decision-Tree)	(Extra-Trees)	(Random-Forest)
MW	0.01	0.36	0.3	0.29
NE	0.3	0.3	0.43	0.44
S	0.03	0.01	0.27	0.25
W	0.34	0.26	0.41	0.39

Figure 11: Binary Classification F1-scores

	(SVM)	(Decision-Tree)	(Extra-Trees)	(Random-Forest)
MW	0.74	0.74	0.74	0.75
NE	0.79	0.79	0.78	0.8
S	0.75	0.75	0.74	0.75
W	0.78	0.8	0.78	0.79

Figure 12: Binary Classification Accuracy Scores

The results from the F1-scores of our binary classification models showed the best model to predict the Mid West region was the decision tree classifier with no hyperparameter tuning. The best model to predict the North East region was the random forest model, which also recorded the highest accuracy. Interestingly, the extra-trees classifier recorded similar performance metrics on the North East region, which indicates ensemble techniques may perform better at predicting desirable linguistic characteristics within this region of the United States. Though ensemble methods are not easily interpretable, they may offer insight into the importance of the linguistic characteristics of these models.

The random forest classifier is one such ensemble that offers a measure of variable importance. Upon further exploratory analysis of our models and their predictions, we can see the 3 most important features of our predictions are the Vader SentimentIntensityAnalyzer, nouns (NN) and adjectives (JJ), *see Figure 13*. An ensemble's feature importance can also be used to see interactions with one another, *see Figure 14*. I leave it to future research to explore these various interactions of linguistic characteristics within these English Varieties.

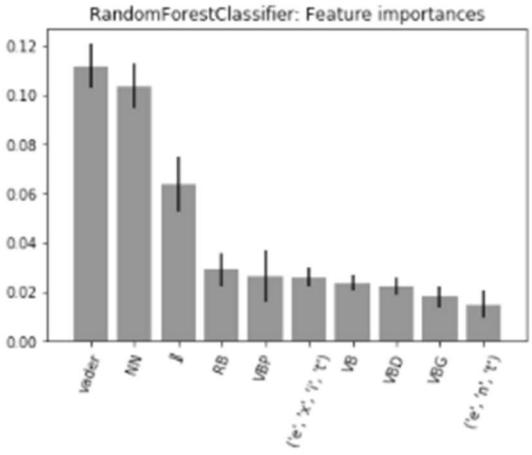


Figure 13: Random Forest Feature Importances

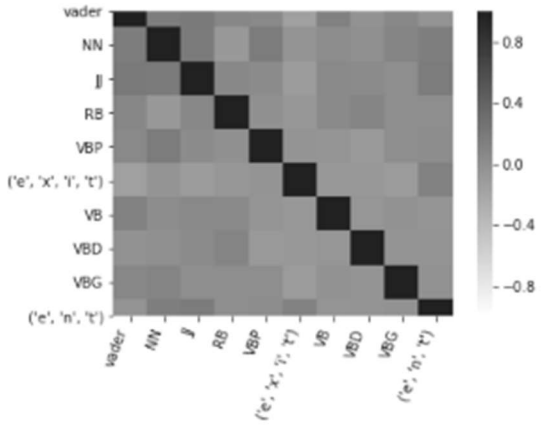


Figure 14: Correlation Matrix of Top 10 Important Features of Random Forest Classifier

The best model to predict both the South and West regions proved to be the extra-trees model. The South region has resulted in the lowest F1-scores among any other region and across all classifiers. The low F1-scores indicate that there exists little discernable linguistic variation among the English Varieties. This is an interesting finding, given that in Part I of this research, all 3 geographical locations outside of the contiguous United States had semantic characteristics most like the South region.

An interesting area of research would be to explore the linguistic characteristics of our South region and their relationship to English linguistic characteristics of tweets from other geographical regions of the world. I leave it to future research to explore these relationships.

29.5 Discussion

This research sheds light on language found on the social media microblogging website, Twitter. Future researchers could consider English text posted on other social media platforms (e.g. Facebook, Instagram, Reddit), to overcome the limitations of single source analysis.

As computational language modeling continues, the interactive effects found in varying forms of textual analysis (social media textual data) will need to be addressed. As the world becomes increasingly automated, textual analysis will become more important to society to deter implicit biases that humans share. It is crucial that computational language modeling reflect all people around the world equally. Biases continue to be found in computational linguistic research, both directly and indirectly [43, 50]. Research, like that presented in this paper, can aid in the prevention of any single English Variety overshadowing another.

This paper does not explore the grammar of the English language in detail. The models presented in this paper are more meaningful in the context of social media analysis because they make predictions of English Varieties as a whole. This allows our models to be more robust and provide more accurate descriptions of evaluation metrics. It should be noted, combining linguistic elements in novel ways produces models that are challenging for any person to understand and explain.

An important question raised by social media analysis, is how to update a model that accounts for language change and how linguistic elements evolved over time. I leave it to future research to explore ways to evaluate varieties and integrate some indication of these linguistic changes over time, and apply these methods more broadly across a larger timeframe with more textual data, from different domains, and different sources.

These findings can be used to identify helpful reviews, deter online bullying, remove harassing or hateful messages from social media platforms, and aid research of on-line social media first responder systems for natural disasters.

This study was not conducted to exploit any particular social media platform or dataset. Twitter textual data is one of the most abundant linguistic sources available to everyone. My hope is that this work motivates future researchers to delve deeper into social media textual analysis in ethical ways and explore these relationships through the lens of linguistic style and other traits of NLP and ML.

30. ACKNOWLEDGMENTS

I thank my advisor, Professor Upal, for his guidance throughout this research. I thank Professor Mansour for all his office hours during my first year. I thank Professor Redmond for teaching me everything data visualization. Most importantly, I thank my wife and family.

31. REFERENCES

[1] Peter Trudgill and Jean Hannah. 2008. International English: A guide to varieties of Standard English. Hodder Education, London, fifth edi-tion.

[2] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[3] S. A. Tagliamonte and D. Denis. 2008. Linguistic ruin? LOL! Instant messanging and teen language. *Ameri- can Speech*, 83.

[4] https://scikit-learn.org/stable/modules/tree.html#tree

[5] Graesser, A. C., Singer, M., & Trabasso, T. (1994). Constructng inferences during narrative text comprehension. Psychological Review, 101, 371-395.

[6] Halliday, M. A. K., & Hasan, R. (1976). Cohesion in English London: Longman.

[7] Trabasso, T., Suh, S., Payton, P., & Jain, R. (1995). Explanatory inferences and other strategies during comprehension and their effects on recall. In R. F. Larch & E. J. O'Brien (Eds.), Sources of Coherence in Reading. Hillsdale, NJ: Erlbaum.

[8] D. Yogatama, C. Dyer, W. Ling, and P. Blunsom, "Generative and Discriminative Text Classification with Recurrent Neural Networks," 2001.

[9] W. Labov. 1966. *The Social Stratification of English in New York City*. Center for Applied Linguistics.

[10] Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

[11] Stephen Roller, Michael Speriosu, Sarat Rallapalli, Benjamin Wing, and Jason Baldridge. 2012. Supervised text-based geolocation using language models on an adaptive grid. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 1500–1510. Jeju Island, Korea.

[12] Benjamin P. Wing and Jason Baldridge. 2011. Simple supervised document geolocation with geodesic grids. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT '11, pages 955–964. Association for Computational Linguistics, Portland, Oregon, USA.

[13] Julian Brooke and Graeme Hirst. 2012. Robust, lexicalized native language identification. In Proceedings of COLING 2012, pages 391– 408. The COLING 2012 Organizing Committee, Mumbai, India.

[14] Adam Kilgarriff. 2001. Comparing corpora. International Journal of Corpus Linguistics, 6(1):97–133.

[15] Evan Sandhaus. 2008. The New York Times An-notated Corpus. Linguistic Data Consortium, Philadelphia, PA.

[16] Lou Burnard. 2000. The British National Corpus Users Reference Guide. Oxford University Computing Services.

[17] Graeme Hirst and Olga Feiguina. 2007. Bigrams of syntactic labels for authorship discrimination of short texts. Literary and Linguistic Computing, 22(4):405–417.

[18] Ying Zhao and Justin Zobel. 2005. Effective and Scalable Authorship Attribution Using Func-tion Words. In Asia Information Retrieval Symposium, pages 174–189. 15.

[19] Bo Han, Paul Cook, and Timothy Baldwin. 2012. Geolocation prediction in social media data by finding location indicative words. In Proceedings of COLING 2012, pages 1045–1062. The COLING 2012 Organizing Committee, Mumbai, India.

[20] Tamaredo, I. (2018). Pronoun omission in high-contact varieties of English Complexity versus efficiency. English World-Wide 39, 85–110. doi: 10.1075/eww.00004.tam

[21] Dunn, J. (2019). *Global Syntactic Variation in Seven Languages: Toward a Computational Dialectology*. *2*(August), 1–22. https://doi.org/10.3389/frai.2019.00015

[22] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. Machine Learning, 20:273–297.

[23] Harris Drucker, Vladimir Vapnik, and Dongui Wu. 1999. Support vector machines for spam categorization. IEEE Transactions on Neural Networks, 10:1048–1054.

[24] Dunn, J. (2018a). Finding variants for construction-based dialectometry a corpus-based approach to regional CxGs. Cogn. Linguist. 29, 275–311. doi: 10.1515/cog-2017-0029

[25] Dunn, J. (2019b). "Modeling global syntactic variation in english using dialect classification," in Proceedings of the NAACL 2019 Sixth Workshop on NLP for Similar Languages, Varieties and Dialects (Minneapolis, MN: Association for Computational Linguistics), 42–53.

[26] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html

[27] Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc V. Sequence to sequence learning with neural networks. In NIPS, 2014.

[28] Graves, Alex. Generating sequences with recurrent neural net- works. arXiv:1308.0850, 2013.

[29] Thorsten Joachims. 1998. Text categorization with support vector machines: learning with many relevant features. In Proceedings of the 10th European Conference on Machine Learning, pages 137–142. Chemnitz, Germany. 13.

[30] Kretzschmar, W. A. (1992). Isoglosses and predictive modeling. Amer. Speech 67, 227–249. doi: 10.2307/455562
[31] Kretzschmar, W. A., Juuso, I., and Bailey, C. (2014). Computer simulation of dialect feature diffusion. J. Linguist. Geogr. 2, 41–57. doi: 10.1017/jlg. 2014.2
[32] Kruger, H., and van Rooy, B. (2018). Register variation in written contact varieties of English A multidimensional analysis. Engl. World-Wide 39, 214– 242.doi: 10.1075/eww.00011.kru
[33] Calle-Martin, J., and Romero-Barranco, J. (2017). Third person present tense markers in some varieties of English. Engl. World-Wide 38, 77–103. doi: 10.1075/eww.38.1.05cal
[34] Grafmiller, J., and Szmrecsanyi, B. (2018). Mapping out particle placement in Englishes around the world A study in comparative sociolinguistic analysis. Lang. Variat. Change 30, 385–412. doi: 10.1017/S09543945180 00170
[35] Tamaredo, I. (2018). Pronoun omission in high-contact varieties of English Complexity versus efficiency. English World-Wide 39, 85–110. doi: 10.1075/eww.00004.tam
[36] Argamon, S., and Koppel, M. (2013). A systemic functional approach to automated authorship analysis. J. Law Policy 12, 299–315.
[37] Hirst, G., and Feiguina, O. (2007). Bigrams of syntactic labels for authorship discrimination of short texts. Liter. Linguist. Comput. 22, 405–417. doi:
[38] Goldberg, Y. (n.d.). *Neural Network Methods for Natural Language Processing*. https://doi.org/10.2200/S00762ED1V01Y201703HLT037
[39] Szmrecsanyi, B., Grafmiller, J., Heller, B., and Rothlisberger, M. (2016). Around the world in three alternations Modeling syntactic variation in varieties of English. *English World-Wide* 37, 109-137. Doi:10.1075/eww.37.2.01.szm
[40] H. AlMubaid, "A learning-classification based approach for word prediction," Int. Arab J. Inf. Technol., vol. 4, no. 3, pp. 264–271, 2007
[41] Cook, P., and Brinton, J. (2017). Building and evaluating web corpora representing national varieties of english. Lang. Resour. Eval. 51, 643–662. doi: 10.1007/s10579-016-9378-z
[42] Rangel, F., Rosso, P., Potthast, M., and Stein, B. (2017). "Overview of the 5th author profiling task at PAN 2017: gender and language variety identification in twitter," in CLEF 2017 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings, vol. 1866. Available online at: https://ceur-ws.org
[43] Jurgens, D., Tsvetkov, Y., and Jurafsy, D. (2017). "Incorporating dialectal variability for socially equitable language identification," in Proceedings of the Annual Meeting for the Association for Computational Linguistics (Vancouver, BC: Association for Computational Linguistics), 51–57.
[44] Cook, P., and Brinton, J. (2017). Building and evaluating web corpora representing national varieties of english. Lang. Resour. Eval. 51, 643–662. doi: 10.1007/s10579-016-9378-z
[45] R. Foulds, "Communication rates of non-speech expression as a function in manual tasks and linguistic constraints," in Proc. Int. Conf. Rehabil. Eng., 1980, pp. 83–87.
[46] M. Ghayoomi and S. Momtazi, "An overview on the existing language models for prediction systems as writing assistant tools," in Proc. IEEE Int. Conf. Syst., Man Cybern., San Antonio, TX, USA, Oct. 11–14, 2009, pp. 5083–5087.
[47] P. Vyrynen, "Perspectives on the utility of linguistic knowledge in english word prediction," Ph.D. dissertation, Univ. Oulu, Linnanmaa, Oulu, Finland, Nov. 19, 2005.
[48] Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *ACL-95*, pp. 189-196.
[49] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
[50] Bolukbasi, T., Chang, K.-W., Zou, J., Saligrama, V., and Kalai, A. (2016). "Debiasing word embedding," in 30th Conference on Neural Information Processing Systems (Barcelona), 1–9.
[51] Jan Rybicki, "Vive La Différence: Tracing the (Authorial) Gender Signal by Multivariate Analysis of Word Frequencies," *Digital Scholarship in the Humanities*, vol. 31, no. 4 (December 2016), pp. 746–61, https://doi.org/10.1093/llc/fqv023.
[52] Sean G. Weidman and James O'Sullivan, "The Limits of Distinctive Words: Re-Evaluating Literature's Gender Marker Debate," *Digital Scholarship in the Humanities*, 2017, https://doi.org/10.1093/llc/fqx017.
[53] Ted Underwood, David Bamman, and Sabrina Lee, "The Transformation of Gender in English-Language Fiction", *Cultural Analytics*, Feb. 13, 2018, DOI: 10.7910/DVN/TEGMGI.
[54] Moshe Koppel, Jonathan Schler, and Shlomo Argamon, "Computational Methods in Authorship Attribution," *Journal of the Association for Information Science and Technology*. vol. 60, no. 1 (January 2009), pp. 9–26, https://doi.org/10.1002/asi.v60:1.

About the authors:

Daniel J. Bahntge is a graduate student at Mercyhurst University.

Cohort Analysis: Personality Prediction Using Deep Learning and Twitter

Cassandra A. Egelston
Department of Computing & Information Science
Mercyhurst University, Erie, PA
Cegels94@Lakers.Mercyhurst.edu

ABSTRACT

Collegiate athlete transfer numbers are reaching an all-time high. Student-athletes transfer for a variety of different reasons, including wrong school choice socially or academically, wrong coach and/or coaching style, lost interest in the sport, injury or lack of performance academically [1]. The objective of this research is to build a deep learning model that can predict a prospective student athlete's personality type using Twitter, in hopes to make the recruiting process more efficient. The dataset used for this research comes from open source platforms.

Keywords

Myers Briggs, personality detection, recruiting, transfers, deep learning, Twitter, Keras.

32. INTRODUCTION

A collegiate athlete's personality can affect the overall success of a team. Therefore, a crucial aspect of a coach's job is recruiting. A coach needs to recruit "student-athletes who fit the needs of the program and improve the team's performance" [4]. While recruiting for the future of the program, it is important to consider how the athlete's personality will fit into the team. "Sports don't build character. They reveal it" [3]. Knowing the personalities of young athletes at the beginning stages will make the recruiting process more efficient.

Currently, coaches have their current players take the Myers Briggs Personality Test in hopes to provide insight on the players' personalities. Personalities have a direct connection on how individuals "work with others, problem-solve and, communicate"[4]. Personalities directly affect athletic performance. Successful athletes work with one another to accomplish a common goal, embrace obstacles, and communicate with each other on and off the field. The goal of this project is to predict a student athletes' personality during the recruitment process. I am interested in solving this problem because throughout my athletic career there have been many circumstances where student athletes did not fit a teams' dynamic. Personality detection during the recruitment process would lessen the misplacement of athletes, which would increase the overall success of the team and the athlete.

As of March 2018, 490,000 student-athletes were participating in a collegiate sport across 19,500 teams, and three divisions [14]. "A 2018 study from the National Student Clearinghouse estimates that 39% of all undergraduates who initially enroll in a four-year institution transfer schools at least once" [8]. Student-athletes transfer for a variety of reasons. According to Pugh [1], student-athletes transfer because they chose the wrong school socially or academically, the wrong coach and or coaching style, lost interest in the sport, got injured, or didn't perform academically [1]. If coaches could use social media to predict prospective athletes' personalities and how their characteristics would fit into the current team, fewer athletes would transfer. Making teams more successful and the recruiting process more efficient and effective for both athletes and their coaches.

This is a data science problem because it can be solved through the collection and analysis of Twitter data. Twitter's social media presence is constantly growing. "Every second, on average, around 6,000 tweets are tweeted on Twitter, which corresponds to over 350,000 tweets sent per minute, 500 million tweets per day, and around 200 billion tweets per year" [15]. The growing social media presence provides adequate data to solve this specific problem.

33. RELEVANT WORK

While, personality prediction is not a new research topic, it has remained largely unexplored in the world of athletics. Past efforts have used social media platforms and the Big Five Personality Test to predict personalities [2][6][12] [16][5][11]. The Myers Briggs personality test is used by psychology researchers to "define personality in terms of five core traits, which can be thought of as stable dispositions that drive behavior" [17]. The five traits consist of openness, conscientiousness, extraversion, agreeableness, and neuroticism. "The five-factor model is used to help understand and predict relationships between personality traits and success in social, academic, and professional circumstances" [17].

Table 2: Big Five Personality Traits Descriptions [18]

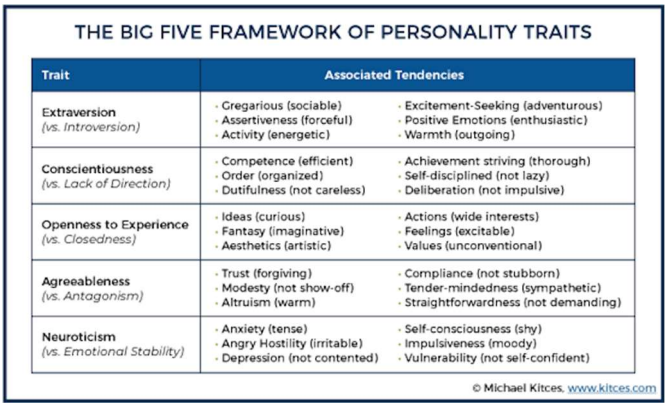


Table 1, illustrates how Myers Briggs defines the personality traits in the Big Five Personality test.

Golbeck *et al.* claimed to be the first to attempt "to bridge the gap between social media and personality research by using the information people reveal in their online profiles" [5]. The authors "core research question asks whether social media profiles can predict personality traits" [5]. The data was collected through a Twitter application. The application "administered a 45-question version of the Big Five Personality Inventory to users" [5]. The researchers collected the most recent 2000 tweets from their fifty subjects. Two main tools were used to analyze a subject's tweets, Linguistic Inquiry Word Count and MRC Psycholinguistic Database, creating a total of 93. They also used the General Inquirer dataset to perform "a word by word sentiment analysis of each user's tweets" [5]. After the data was collected, they examined the correlations between personality and twitter behavior. The authors ran "a Pearson correlation analysis between subjects' personality scores and each of the features obtained from analyzing their tweets and public account data"[5].

Table 3: Results from Golbeck *et al.* study.

Language Feature	Examples	Extro.	Agree.	Consc.	Neuro.	Open.
"You"	(you, your, thou)	0.068	**0.364**	**0.252**	-0.212	-0.020
Articles	(a, an, the)	-0.039	-0.139	-0.071	-0.154	**0.396**
Auxiliary Verbs	(am, will, have)	0.033	0.042	**-0.284**	0.017	0.045
Future Tense	(will, gonna)	0.227	-0.100	**-0.286**	0.118	0.142
Negations	(no, not, never)	-0.020	0.048	**-0.374**	0.081	0.040
Quantifiers	(few, many, much)	-0.002	-0.057	-0.089	-0.051	**0.238**
Social Processes	(mate, talk, they, child)	**0.262**	0.156	0.168	-0.141	0.084
Family	(daughter, husband, aunt)	**0.338**	0.020	-0.126	0.096	0.215
Humans	(adult, baby, boy)	0.204	-0.011	0.055	-0.113	**0.251**
Negative Emotions	(hurt, ugly, nasty)	0.054	-0.111	**-0.268**	0.120	0.010
Sadness	(crying, grief, sad)	0.154	-0.203	**-0.253**	0.230	-0.111
Cognitive Mechanisms	(cause, know, ought)	-0.008	-0.089	**-0.244**	0.025	0.140
Causation	(because, effect, hence)	0.224	**-0.258**	-0.155	-0.004	**0.264**
Discrepancy	(should, would, could)	0.227	-0.055	**-0.292**	0.187	0.103
Certainty	(always, never)	0.112	-0.117	-0.069	-0.074	**0.347**
Perceptual Processes						
Hearing	(listen, hearing)	0.042	-0.041	0.014	**0.335**	-0.084
Feeling	(feels, touch)	0.097	-0.127	**-0.236**	**0.244**	0.005
Biological Processes	(eat, blood, pain)	-0.066	0.206	0.005	0.057	**-0.239**
Body	(cheek, hands, spit)	0.031	0.083	-0.079	0.122	**-0.299**
Health	(clinic, flu, pill)	**-0.277**	0.164	0.059	-0.012	-0.004
Ingestion	(dish, eat, pizza)	-0.105	**0.247**	0.013	-0.058	-0.202
Work	(job, majors, xerox)	0.231	-0.096	**0.330**	-0.125	**0.426**
Achievement	(earn, hero, win)	-0.005	**-0.240**	-0.198	-0.070	0.008
Money	(audit, cash, owe)	-0.063	**-0.259**	0.099	-0.074	0.222
Religion	(altar, church, mosque)	-0.152	-0.151	-0.025	**0.383**	-0.073
Death	(bury, coffin, kill)	-0.001	0.064	**-0.332**	-0.054	0.120
Fillers	(blah, imean, youknow)	0.099	-0.186	**-0.272**	0.080	0.120
Punctuation						
Commas		0.148	0.080	**-0.24**	0.155	0.170
Colons		-0.216	-0.153	**0.322**	-0.015	-0.142
Question Marks		**0.263**	-0.050	0.024	0.153	-0.114
Exclamation Marks		-0.021	-0.025	**0.260**	**0.317**	**-0.295**
Parentheses		**-0.254**	-0.048	-0.084	0.133	**-0.302**
Non-LIWC Features						
GI Sentiment		0.177	-0.130	-0.084	-0.197	**0.268**
Number of Hashtags		0.066	-0.044	-0.030	-0.217	**-0.268**
Words per tweet		**0.285**	-0.065	-0.144	0.031	0.200
Links per tweet		-0.061	-0.081	**0.256**	-0.054	0.064

Table 3 shows the results of the Pearson Correlation analysis.

The authors performed a regression analysis using Weka to "predict the score of a given personality feature. Two regression algorithms: Gaussian Process and ZeroR, each with 10-fold cross-validation with 10 iterations were used Two algorithms had similar performance over the personality features" [5]. The analysis enabled the authors to predict personality scores within 11% - 18% of their actual values.

Sumner *et al.* "sought to examine the relationship between Dark Triad personality traits and Twitter activity and examine whether machine learning could be used to predict these constructs based solely on Twitter usage" [11]. Dark Traid personalities consist of narcissism, Machiavellianism, and psychopathy. The study was performed on "2,927 Twitter users from 89 countries" [11]. A maximum of 3,200 tweets was collected using Twitter API. The tweets were analyzed using Linguistic Inquiry and Word Count (LIWC), resulting in 337 features. The following models were used to predict the correlation between Dark Traid personality traits and Twitter activity.

1) "Support Vector Machine (SVM) using sequential minimal optimization (SMO) and a polynomial kernel" [11]
2) "Random Forest, an ensemble method that combines multiple decision trees" [5]
3) "J48, an implementation of the C4.5 decision tree algorithm" [11]
4) "Naïve Bayes (NB) classifier" [11]

Summer *et al.* used several models from Kaggle, which is an online data science platform, to complement the following algorithms. The results of the research "identify several statistically significant correlations between Dark Personality traits and Twitter usage" [11]. The study showed that people who use profanity or words associated with anger while tweeting tend to have higher scores in psychopathy and Machiavellianism. Narcissism correlated the number of followers and friends a user had. "People with higher scores in narcissism tended to have both more followers and friends and a greater number of followers per friend. Narcissism was also positively correlated with Klout scores, a score associated with exerting influence over other users through online behavior" [11]. The study shows that there is a correlation between the Big Five Personality Traits and the Dark Traid. However, the "practical performance of machine prediction is currently poor when applied directly to an individual" [11].

Quercia *et al.* examined the relationship between personality scores and Twitter users. The research project looked at Twitter's publicly available information, consisting of a user's followers, following, and listed. Using this information, they were able to identify three types of users, including listeners, popular, and highly readable. The authors also identified influential users, using two scores "Klout" and "TIME"[7]. The "Klout" score doesn't "consider the number of followers or tweets instead, it considers whether a user's tweets are clicked, replied, and further propagated (retweeted)"[7]. "TIME" is a measurement that is generated by "TIME magazine to rank public figures such as Barack Obama, Oprah Winfrey, and Lady Gaga. The measure combines one's popularity on both Twitter and Facebook by computing 2·nfollowers+nfacebook·2, where nfollowers is the number of Twitter followers, and nfacebook is the number of Facebook social contacts"[7]. The project studies the "product-moment correlation between the logarithm of the five user characteristics and each of the (big) five personality traits, plus two additional attributes, namely age and sex"[7].

The data is collected through a "Facebook application called myPersonality"[7]. This application is used to detect the relationship between personality scores and Twitter users. "The application ensures high test result validity by removing the protocols that may be a product of inattentive, language incompetent, or randomly responding individuals" [7]. The study analyzed 335 Twitter users. All the users analyzed "listed their Twitter account on their Facebook profile" [6]. The following machine learning techniques were used to predict personalities.

1) Region analysis
2) 10-fold cross-validation with 10 iterations and te M50 Rules
3) Root mean squared error

The results of the research identified a correlation between personality scores, big five personality traits, and the five user characteristics. The strongest correlation was found between listeners and popularity associated with extroversion and Neuroticism. Extraversion had a score of "0.13 for listeners and

Trait	Listeners *log(Following)*	Popular *log(Followers)*	Highly-read *log(Listed)*	Influential *Klout*	Influential *log(TIME)*
O	0.05	0.05	**0.17***	0.13	0.00
C	0.08	0.10	0.02	0.01	**0.18*****
E	**0.13***	**0.15****	0.09	**0.15***	**0.25*****
A	0.07	0.02	0.03	-0.17	0.06
N	**-0.17****	**-0.19*****	-0.03	**-0.03***	**-0.20*****
log(Age)	**0.28***	**0.37***	0.13	0.05	**0.39***
Male	-0.05	-0.05	-0.05	-0.04	0.01

0.15 for popular users. Neuroticism had a score of −0.17 for listeners and −0.19 for popular users"[7].

Table 4: Results from Quercia *et al.* study [7].

Table 4 shows the "correlation coefficients between big five personality traits and five quantities that characterize listeners, popular users, highly-read users, and (Klout & time) influential users. Statistically significant correlations are in bold and their p-values are expressed with *'s: $p < 0.001$ (***), $p < 0.01$ (**), and $p < 0.05$ (*)" [7]. The authors concluded that there are two key takeaways from the study. First takeaway: Twitter user types show a variety of similarities and differences. "All user types (listeners, popular, highly-read, and influential users) are emotionally stable (low in Neuroticism), and most of them are extrovert. These inferences have long been supported informally by intuition but have been difficult to make precise. Interestingly, popular users tend to be 'imaginative', while influential users tend to be organized"[7]. Second takeaway: "user personality can be easily and effectively predicted from public data, and that suggests future directions in a variety of areas"[7]. Quercia *et al.* believe personality predictions can be useful in the following areas: marketing, user interface design, and recommender systems.

34. PROPOSED SOLUTION

The training dataset was collected through Kaggle, which is an online data science platform. The data set consists of 8,600 rows of data. Each row of data contains a "persons 4 letter MBTI code/type and the last 50 things they have posted" [19]. Each post is separated by (|||). The personality type data "was collected through the PersonalityCafe forum, as it provides a large selection of people and their MBTI personality type, as well as what they have written" [20]. After collecting the data set, I put the data into Qlik, which is a data visualization tool, to see how the data was distributed by personality type. The following pie chart shows the imbalance within the personality data set.

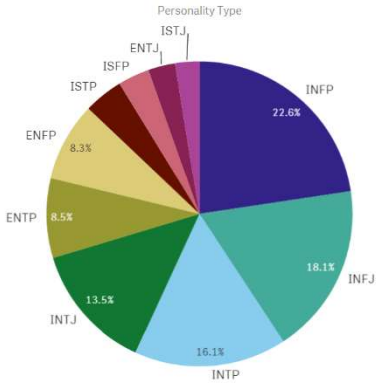


Figure 1: Personality Data Set Distribution

The testing data was collected through Twitter API, using the tweepy python package. I used usernames to collect the last 300 tweets of the current Mercyhurst softball players. "Twitter is built on the conversations happening around the world via Tweets. With Twitter's API platform, you will find endpoints to unlock the data from Tweets, so you can build great experiences and solutions for your customers. These endpoints enable you to manage your Tweets, publish and curate Tweets, filter and search for Tweet topics or trends, and much more" [21].

Before reading in the training data, I balanced the data by dividing the highest personality form the other personality types and duplicating the difference. Once the data was balanced, I calculated the following features, the number of question marks, examination points, at symbols, hashtags, pictures, videos, music, HTML links, and words per post. After the metrics were counted, I performed a pre-text analysis, converting all text to lower case, removing numbers, punctuations, stop words, hashtags, HTML. I also implemented stemming, which returned the words in their root form. I used tokenization to turn each text into a sequence of integers, setting the max number of words to 500. I then used padding to make the sequences to the same length of 400. I then randomly split data into train/validation sets and used label encoder to encode target labels. After implementing future engineering, I used Keras to implement a sequential model, using Adam and SGD as optimizers. I chose this model because its multi-layer neural networks allow it to learn extremely complex patterns, such as personality types [6]. The model has five layers. The first layer of the model is, model.add(Dense(128, input_shape=(408,))). The dense layer allows the model to output arrays with the shape of (*,128). The input shape allows the model to take arrays with the shape of (*,408) as the input. The input layer is flattened before the initial dot product because it has a rank greater than 2 [22]. "Flattening transforms a two-dimensional matrix of features into a vector that can be fed into a fully connected neural network classifier"[23]. The second layer of the model is, model.add(LeakyReLU(alpha=0.05)). Leaky ReLU is a nonlinear activation function. I used leaky ReLU instead of ReLU because it eliminates the "dying ReLU" problem. The "dying ReLU" problem occurs when "inputs approach zero, or are negative, the gradient of the function becomes zero, the network cannot perform backpropagation and cannot learn" [24]. Leaky ReLU eliminates this issue by not reducing the negative part to 0, it rather divides the negative part by a large value. Alpha defines the slope of the curve, where x is less than 0 [13]. The third layer is,

model.add(Dropout(0.5)). A dropout layer helps prevent overfitting, by "randomly setting a fraction rate of inputs to 0 at each update during training" [22]. The fourth layer is, model.add(Dense(2, activation='sigmoid')). Sigmoid is a nonlinear activation function. This function prevents jumps in the output values and normalizes the output of each neuron (between 1 and 0) [24]. The last layer is, model.add(Dense(2, activation='softmax')). Softmax is a nonlinear activation function. I implemented the softmax activation function because it can handle multiple classes, which fits the problem I am trying to solve as there are 16 different personality types.

I used the following hyperparameters to get the best results:

1) Learning_rate = 0.001
2) batch_size=100
3) epochs=70
4) validation_split=0.1

35. EVALUATION

After implementing my sequential classification model, I evaluated the model using an accuracy score and categorical crossentropy loss function. Accuracy score "computes subset accuracy: the set of labels predicted for a sample must exactly match the corresponding set of labels in y_true" [25]. "Categorical crossentropy is a loss function that is used for single label categorization. This is when only one category is applicable for each data point. In other words, an example can belong to one class only" [26].

$$L(y,\hat{y}) = -\sum_{j=0}^{M}\sum_{i=0}^{N}(y_{ij} * log(\hat{y}_{ij}))$$

Figure 2: Categorical Crossentropy Math [26]

The loss ratio compares "the predictions (the activations in the output layer, one for each class) with the true distribution, where the probability of the true class is set to 1 and 0 for the other classes" [26]. I furthered my evaluation by having my study participants take the Myers Briggs Personality Test, which assigns the following personality type to the test taker.

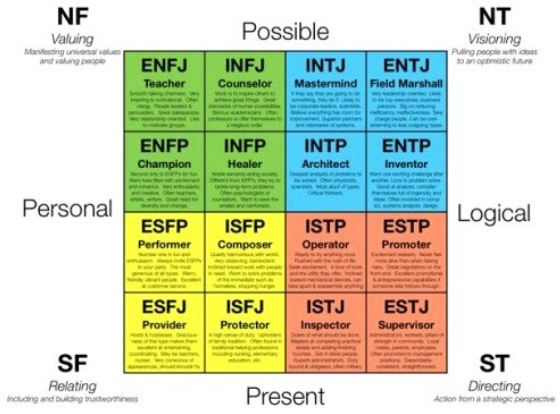


Figure 3: Categorical Crossentropy Math [26]

I compared the results of the actual test to the personality type I was predicting, to validate my results.

36. RESULTS

The goal of a learning algorithm is to create a good fitting curve, which exists between over and under the fitted model. The algorithm produces a good fitting model around 70 epochs, which is where my model converged, producing the best results. As the training rates increase in epoch, we see better results. The training loss decreases to a point of stability creating a generalization gap between the two final loss values. The model predicted introversion and extroversion labels with accuracies as high as 97% and low of 78%. The model predicted sensing and intuition labels with accuracies as high as 99% and low of 78%. The model predicted judging and perceiving labels with accuracies as high as 95% and low of 75%. The model predicted thinking and feeling labels with accuracies as high as 95% and low of 72%. The training accuracy is lower than the testing accuracy because the model I created has a drop out of .05. After 30 epochs the model began to overfit the data, causing the testing and training lines to split, creating a larger generalization gap.

Table 4: Categorical accuracy results

Accuarcy	I-E	I-S	J-P	T-F
Low	78%	78%	75%	72%
High	97%	99%	95%	95%

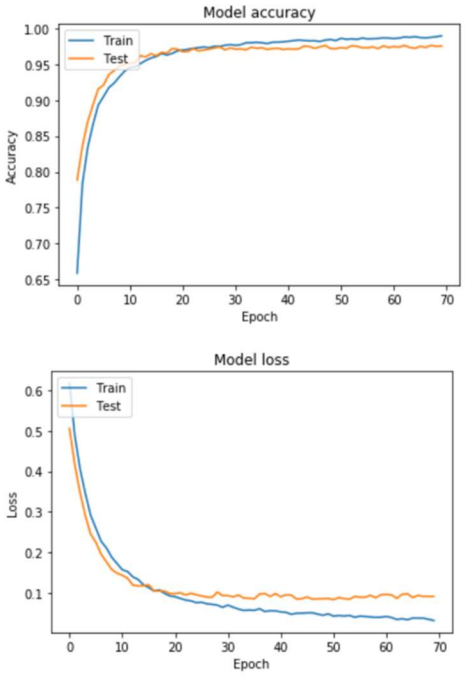


Figure 3: Categorical accuracy & Categorical Cross Entropy Loss (Train vs Test, 70 Epoch)

Although the learning algorithm produces a good fitting curve, it predicts inaccurate personality types when compared to the participants Myers Briggs Personality Test results. The test compares four areas of preferences, which create sixteen personality types. The area of preferences consists of:

1) Extraversion vs introversion

a. "An extravert's source and direction of energy expression is mainly in the external world, while am introvert has a source of energy mainly in their own internal world" [27].

2) Sensing vs intuition
   a. "Sensing means that a person mainly believes information he or she received directly from the external world. Intuition means that a person believes mainly information he or she receives from the internal or imaginative world" [27].
3) Thinking vs feeling
   a. "Thinking means that a person decides mainly through logic. Feeling means that, as a rule he or she makes a decision based on emotion" [27].
4) Judging vs perceiving.
   a. Judging means that a person organized all his life events and as a rule sticks to his plans. Perceiving means that he or she is inclined to improvise or explore alternative options.

The model has a hard time detecting the difference between introversion vs extroversion, judging vs perceiving, thinking vs feeling. The model predicts that someone is an extrovert 100% of the time. I believe it would be able to detect the difference between introversion and extroversion if we considered features such as how many followers/ following a person has or how many of their tweets contain emojis. The model was 93% accurate when predicting judging vs perceiving. I believe we could improve the accuracy score for this prediction if we considered the number of retweets and favorites an individual has. I believe an individual who is perceiving is more likely to retweet or favor someone else tweet. The model predicts that someone is feeling 100% of the time. I believe the model would better detect the difference between feeling and thinking if we considered how many emojis someone uses when tweeting. A feeling individual bases their decisions off emotions, so they would be more likely to express their feeling through emojis. The model was 100% accurate when detecting the difference between judging and perceiving.

Table 5: Results from Cohort Analysis: Predicting Personality Characteristics Using NLP and Twitter Study

Participant	Myers Briggs Test Results	I/E	S/I	T/F	J/P
Participant 1	ENFJ	E	N	F	J
Participant 2	ENFJ	E	N	F	J
Participant 3	ENFJ	E	N	F	J
Participant 4	ENFJ	E	N	F	J
Participant 5	ESFJ	E	S	F	J
Participant 6	ENFJ	E	N	F	J
Participant 7	ISFJ	E	S	F	J
Participant 8	ENFP	E	N	F	J
Participant 9	ENFJ	E	N	F	J
Participant 10	ENTJ	E	N	F	J
Participant 11	ENFJ	E	N	F	J
Participant 12	ESFP	E	S	F	P
Participant 13	ESTJ	E	S	F	J
Participant 14	ENTP	E	N	F	P
Participant 15	ENFP	E	N	F	P
Participant 16	ESFJ	E	S	F	J

Key:
Myers Briggs Personality Test Results (Navy Blue)
Model Prediction Results (Gray)
Wrongly Predicted Results (Red)

The personality predictions are similar when compared to the Myers Briggs Test Results however, they are not the same. The model has a hard time detecting the difference between introversion vs extroversion, judging vs perceiving, thinking vs feeling. Why did the results differ in these preference areas? The online disinhibition effect causes individuals to "say and do things in cyberspace that they wouldn't ordinarily say and do in the face-to-face world. They loosen up, feel less restrained, and express themselves more openly" [9]. There are two types of online disinhibition, benign and toxic disinhibition. Benign disinhibition is when people use the cyber world to share secrets, emotions, fear, and show unusual acts of kindness. Toxic distribution is the complete opposite, it is when people use rude language, criticism, anger, crime, and violence to express themselves. Online environments give people a sense of invisibility, which "gives people the courage to go places and do things that they otherwise wouldn't" [10]. I believe the model's predictions and the results from the Myers Briggs personality test differed because of the online disinhibition effect.

37. CONCLUSION

The model predicted introversion and extroversion labels with accuracies as high as 97% and low of 78%. The model predicted sensing and intuition labels with accuracies as high as 99% and low of 78%. The model predicted judging and perceiving labels with accuracies as high as 95% and low of 75%. The model predicted thinking and feeling labels with accuracies as high as 95% and low of 72%. Despite good accuracy scores the model produced, the predictions were inaccurate in the following areas when compared to the Myers Briggs Personality Test results of the participants.

1) Introversion vs extroversion
2) Judging vs perceiving
3) Thinking vs feeling

This could be attributed to the online disinhibition effect.

38. FUTURE WORK

In the future, the model can be trained on real-time twitter data, instead of the Kaggle dataset. This will allow the researcher to incorporate other factors such as the number of followers, following, likes, retweets, pictures, videos, and emojis used. The researcher can connect the emojis used to the personality type by using sentiment analysis. Sentiment analysis will allow the researcher to understand how an individual is feeling based on the emojis they are using, which could better detect the difference between feeling and thinking. The researcher could correlate the number of retweets/favorites to better detect perceiving vs judging. Lastly, the researcher could use the number of followers/following to better detect whether an individual is an extrovert or an introvert.

39. REFERENCES

[1] U. S. Sports Academy. 2016. Factors That Influence Collegiate Student-Athletes to Transfer,

Consider Transferring, or Not Transfer. *The Sport Journal*. Retrieved October 16, 2019 from https://thesportjournal.org/article/factors-that-influence-collegiate-student-athletes-to-transfer-consider-transferring-or-not-transfer/

[2] Mounica Arroju, Aftab Hassan, and Golnoosh Farnadi. Age, Gender and Personality Recognition using Tweets in a Multilingual Setting. 9.

[3] Bob Dyar. 2018. Sports don't build character. They reveal it. *CEDE Sports*. Retrieved October 15, 2019 from https://cedesports.org/sports-dont-build-character-they-reveal-it-2/

[4] Kaitlin S. Fost. 2014. Can a Student-Athlete's Personality Type Affect her Overall Athletic Success? DOI:https://doi.org/10.13016/M29M79

[5] Jennifer Golbeck, Cristina Robles, Michon Edmondson, and Karen Turner. 2011. Predicting Personality from Twitter. In *2011 IEEE Third Int'l Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third Int'l Conference on Social Computing*, IEEE, Boston, MA, USA, 149–156. DOI:https://doi.org/10.1109/PASSAT/SocialCom.2011.33

[6] Gavril Ognjanovski. 2019. Everything you need to know about Neural Networks and Backpropagation — Machine Learning Made Easy…. *Medium*. Retrieved April 23, 2020 from https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-learning-made-easy-e5285bc2be3a

[7] Daniele Quercia, Michal Kosinski, David Stillwell, and Jon Crowcroft. 2011. Our Twitter Profiles, Our Selves: Predicting Personality with Twitter. In *2011 IEEE Third Int'l Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third Int'l Conference on Social Computing*, IEEE, Boston, MA, USA, 180–185. DOI:https://doi.org/10.1109/PASSAT/SocialCom.2011.26

[8] smeyers@ncaa.org. 2019. Research on Student-Athlete Transfers. *NCAA.org - The Official Site of the NCAA*. Retrieved October 16, 2019 from http://www.ncaa.org/about/resources/research-student-athlete-transfers

[9] John Suler. The Online Disinhibition Effect. 6.

[10] John Suler. The Online Disinhibition Effect. 6.

[11] Chris Sumner, Alison Byers, Rachel Boochever, and Gregory J Park. 2012. Predicting Dark Triad Personality Traits from Twitter usage and a linguistic analysis of Tweets. *th International Conference on Machine Learning and Applications ICMLA* (2012), 8.

[12] Tommy Tandera, Hendro, Derwin Suhartono, Rini Wongso, and Yen Lina Prasetio. 2017. Personality Prediction System from Facebook Users. *Procedia Computer Science* 116, (2017), 604–611. DOI:https://doi.org/10.1016/j.procs.2017.10.016

[13] 2019. Using Leaky ReLU with Keras. *MachineCurve*. Retrieved April 21, 2020 from https://www.machinecurve.com/index.php/2019/11/12/using-leaky-relu-with-keras/

[14] Recruiting Fact Sheet WEB.pdf. Retrieved October 16, 2019 from https://www.ncaa.org/sites/default/files/Recruiting%20Fact%20Sheet%20WEB.pdf

[15] Twitter Usage Statistics - Internet Live Stats. Retrieved October 16, 2019 from https://www.internetlivestats.com/twitter-statistics/

[16] Personality Types Prediction based on Machine Learning | Fayrix. Retrieved October 15, 2019 from https://fayrix.com/credit-scoring

[17] Big 5 Personality Traits. *Psychology Today*. Retrieved November 8, 2019 from https://www.psychologytoday.com/basics/big-5-personality-traits

[18] Big-Five-Framework-of-Personality-Traits. *FP Advance*. Retrieved November 19, 2019 from

https://fpadvance.com/business-must-reads-feb-2019/big-five-framework-of-personality-traits/

[19] (MBTI) Myers-Briggs Personality Type Dataset. Retrieved December 4, 2019 from https://kaggle.com/datasnaek/mbti-type

[20] BYO Tweets: Predict your Myers-Briggs Personality. Retrieved December 5, 2019 from https://kaggle.com/stefanbergstein/byo-tweets-predict-your-myers-briggs-personality

[21] Tweets – Twitter Developers. Retrieved December 3, 2019 from https://developer.twitter.com/en/products/tweets

[22] Core Layers - Keras Documentation. Retrieved April 21, 2020 from https://keras.io/layers/core/

[23] Using the Keras Flatten Operation in CNN Models with Code Examples. *MissingLink.ai*. Retrieved April 21, 2020 from https://missinglink.ai/guides/keras/using-keras-flatten-operation-cnn-models-code-examples/

[24] 7 Types of Activation Functions in Neural Networks: How to Choose? *MissingLink.ai*. Retrieved April 21, 2020 from https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/

[25] sklearn.metrics.accuracy_score — scikit-learn 0.22.2 documentation. Retrieved April 4, 2020 from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html

[26] Categorical crossentropy loss function | Peltarion Platform. *Peltarion.com*. Retrieved March 18, 2020 from https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/categorical-crossentropy

[27] Personality Type Explained. Retrieved April 4, 2020 from http://www.humanmetrics.com/personality/type

Predicting the Level of Violence and Participation During Social Unrest with the Use of ELMo and Social Media Language

Alicja Mincewicz
Department of Computing &
Information Science
Mercyhurst University
501 East 38th St
Erie, PA 16546
alicja.mincewicz@gmail.com

Abstract

This research focuses on developing a model that can predict the level of violence and participation during a social event, such as a protest, rally, or demonstration based on Twitter language. The data consists of tweets collected around specific demonstrations throughout the USA, Australia, and the UK. Additional data on the specifics of the events were collected from official news outlets. I utilized Embedded Language Modeling (ELMo) to create a methodology to define the level of violence and participation. In this method, the word representations are functions of the entire input sentence and are computed on top of a deep bidirectional language model (biLM). ELMo considers not only word vectors but also syntax, semantics, and model polysemy.

Keywords

Protest, Demonstration, Violence, Participation, Twitter, Natural Language Processing, Embedded Language Modeling, Text Mining, Social Media Analysis, Social Unrest

40. INTRODUCTION

Many citizens partake in demonstrations to exercise their democratic rights to do so and to express their opinions. In some places, it is a way to initiate change and show approval or disapproval of a socio-economical or geopolitical situation, even if such social events are not necessarily approved by the governments. Approximately 75% of the demonstrations are legal and organized in advance [2]. Additionally, protests with larger attendance are much more likely to be successful and carry a lower risk of arrests. Therefore, participants try to plan and announce demonstrations ahead of time in order to mobilize a larger number of participants and increase their event's success rate [2][10]. In some instances, demonstrations and protests can escalate into violence, causing deaths, injuries, and property damage.

Increasingly, social media serves as one of the main open sources of information and plays an important role in the organization of social events, such as demonstrations, parades, and other gatherings. Twitter alone attracts 126 million active daily users and 321 million active monthly users [3]. Ordinary citizens, government agencies, and journalists use Twitter to disseminate information in real-time, being much faster than traditional news outlets. Social media was widely used during the Baltimore protests in 2015, Arab Springs between 2010 and 2012, and the Charlottesville, VA rally in August 2017 [4][13]. Attendees used Twitter to update others on the developments of the events, to communicate with other participants, and to make further plans. Such events can be costly in many ways, as all three of the above-mentioned protests caused property damage, injuries, and deaths.

Timely and accurate predictions of protests and demonstrations, and the likelihood of violence can help law enforcement, authorities, and public officials to prepare in advance and take necessary precautions. These precautions can not only limit the monetary costs of damage but also protect people's lives. In corporate security, many organizations, such as Amazon, Microsoft, and Facebook have their global security departments and intelligence teams that monitor social and unrest daily. For many organizations, such tasks are done manually by analysts who keep track of many different information outlets using several different types of software. Analysts determine the likelihood of demonstrations escalating into violence with the use of their expertise and other information, such as perceived levels of participation, presence of extremist groups, location, and cause of the demonstration. While some protests are inherently more likely to escalate into violence, leadership bases their actions on the analysts' analysis.

The main objective of this research is to develop a model that estimates the level of violence and the number of participants during a demonstration, protest, or social unrest. Corporate security and their intelligence departments would benefit from the model as it would reduce the amount of manual work the analysts need to perform. Additionally, with the use of Natural Language Processing, the model would pick up on linguistic nuances that a human might not be able to. In this project, I will focus on Embedded Language Modeling with the use of Twitter. The data includes tweets collected around the Unite the Right Rally in August 2017 in Charlottesville, VA, and tweets collected between January and March 2020.

Most of the previous studies have been conducted at the strategic level, such as considering demonstrations and protests as organized social movements and looking at how socioeconomic and political situations impact protest moods [8]. This research

focuses more on the tactical level of analysis and focuses on the estimated number of participants and the level of violence.

41. RELEVANT WORK

This research is related to the following research areas: prediction of events and social behaviors, social media mining, and research on behaviors and language that help predict the level of violence and estimated attendance during an event. Previous studies have mainly focused on event detection and the socioeconomic and geopolitical aspects, while modeling demonstrations as social movements [12]. Scholars have also examined the role of collective identity and the sense of empowerment that participation in protests bring. Groups that are perceived to be strong are more likely to experience anger and are more determined to take action [12]. Combining social sciences, such as psychology and sociology, with computer science allowed researchers to look at modeling behaviors, predicting actions, and predicting participants' behaviors and attitudes [7][8].

Many researchers have investigated predicting social unrest on social media with the use of several methods, with clustering being one of the most prevalent. The existence of the event can be deducted from changes in trends and topic popularity on social media. With the use of cluster analysis, scholars divided the data into subtopics based on common keywords and hashtags [8], creating clusters of different sizes. Additionally, clusters are created dynamically and evolve over time, which enables the model to add new points to the clusters [2][8]. The main limitation of this method is that in large social media networks and datasets, the number of nodes may be too large to use efficiently. To combat this issue, Aggarwal and Subbian [2] used a sketch-based retrieval technique in addition to regular cluster analysis. With the use of this method, the researchers maintained node counts in underlying clusters. Additionally, a sketch table was used to maintain frequency counts of the nodes in the incoming data. With the increase of the sketch-table length, the purity of clusters increases as well, improving the overall accuracy [2]. Aggarwal and Subbian used two datasets for their research: a collection of 1,628,779 tweets and an Enron email dataset.

Other scholars focused more on NLP. Mutial et. al. [10] first used linguistic processing with the use of tokenization, lemmatization, and named entity extraction to analyze the language. The documents were then filtered by phrases and parsed with the use of a dependency parser. The researchers used Probabilistic Soft Logic (PSL) in the geocoding of the news and blog sources. PSL is a framework that uses collective, probabilistic reasoning that uses first-order logic rules [6]. Tweets and Facebook posts that did not already include information on locations and geotags were geotagged based on the locality that the text was about. The aforementioned methods, together with additional rules and constraints to the model, were used to determine dates and locations of demonstrations. This system has been used for several years by analysts in Latin America [6].

Won et. al. researched perceived violence during social unrest with the use of image analysis of a UCLA Protest Image Dataset [14]. The scholars focused on visual analysis with the use of a Convolutional Neural Network (CNN) and OpenFace models. The CNN model uses the input of thousands of single pictures, automatically classifying these images, and outputs a series of prediction scores that included visual attributes, binary image categories (e.g. non-protest and protest), perceived violence, and image sentiment [14]. The OpenFace model with the use of a CelebA facial attribute dataset outputs information on race and gender, among other expressions. While this research is one of very few that address the issue of perceived violence, it does so with image and not text analysis.

This paper takes social media analysis and text mining a step further as it focuses on tactical data on predicting estimated attendance and level of violence. Unlike the aforementioned scholars, this research uses Embedded Language Modeling (ELMo) [11]. The main dataset used for this project includes Tweets with #charlottesville spanning a few days in August 2017. The instance of Charlottesville, VA is a great example of how a spontaneous demonstration can drastically escalate into violence. Additional data was collected between January and March 2020. It includes information on numerous demonstrations that occurred within this timeframe.

42. PROPOSED SOLUTION

To solve the research problem, I used Embedded Language Modeling (ELMo) [11]. I collected data about specific demonstrations and protests by using geolocations and hashtags related to various topics. The Tweet data was cleaned by removing hyperlinks, emoticons, stop words, non-Latin characters, punctuations, and user mentions. Each of the demonstrations, plus tweets on negative data, were treated separately. Each demonstration's dataset included a collection of tweets about that event. That text was compared against the official data on attendance and levels of violence. The language from these tweets was analyzed using Embedded Language Modeling. This model is often used in Sentiment Analysis or Named Entity Extraction [11] as it looks at sentences as a whole to calculate word embeddings. This approach allows for future prediction of perceived attendance and violence based on language changes.

43. METHODOLOGY

43.1 Data

I obtained an existing data set containing tweets posted on 15 August 2017 from Charlottesville, VA [1]. The Tweets were collected following violent demonstrations in that city. Additionally, I collected tweets between January and March 2020 from various locations throughout the US, including Seattle, New York City, Austin, Washington D.C., and Portland. I also collected data from the UK and Australia. These tweets were mined around specific demonstrations. The data also included randomly collected tweets from Los Angeles and Seattle to include negative data. Overall, I collected 28,338 tweets, which were cleaned. Afterward, I looked at official news articles and government websites to obtain the number of participants and determine the level of violence. The number of participants and the level of violence was divided into the below-listed subcategories.

Levels of violence are divided into the following categories (police actions include the use of batons, tear gas, etc.):

- *Level 0* ->–Not a social event.
- *Level 1* -> Low level of violence – no violence during the event/peaceful event.
- *Level 2* -> Medium level of violence – possible injuries, police actions, and property damage.
- *Level 3* -> High level of violence – at least 1 death, police actions, and property damage.

Attendance is divided into the following categories:

- *Level 0* -> 0 participants – not a social event
- *Level 1* -> 1 – 1,000 participants
- *Level 2* -> 1,000 – 10,000 participants
- *Level 3* -> 10,000 – 50, 000 participants
- *Level 4* -> 50,000 participants <

Attendance and violence levels were treated as two separate categories. I first looked at the tweets and levels of violence. The collected data was classified into one of the above categories based on the additional information obtained from news, government data, etc.). Initially, each demonstration/protest was labeled as a whole (e.g. Washington, DC, demonstration – violence level 1) and then divided into individual tweets with the same labels as the event overall. Afterward, I joined all 28,338 tweets and created one Pandas data frame.

	Tweet	Violence_Level
16629	Charlotte Pence Bond author and the daughter ...	1
7082	In solidarity with everyone who protested ton...	1
1051	Fake things like trend overnight because the ...	2
13970	Had great afternoon and performing in Whitehal...	1
1620	The anti Semitic attacks happening across our ...	1
...	...	...
13029	We are fighting for climate justice We are fi...	1
342	ANTIFADomesticTerrorism	2
5766	After that disturbing interview with Morrison ...	1
10738	Can barely capture the whole crowd Huge clima...	1
7682	Epic crowd now heading down Park Street towar...	1

28338 rows × 2 columns

Figure 9: Data Frame for Violence Levels

I later divided the data frame into three parts: training data, validation data, and testing data. The testing data consisted of 2,338 tweets. The training data consisted of 20,800 tweets and the validation data was 5,200 tweets (80% train data and 20% validation data). I implemented the same process to the number of participants.

	Tweet	Participation_Level
10684	Speeches over packed crowds funnel out to beg...	3
880	Fixed	4
7972	Incredible photo taken at the Sydney rally	3
1543	Why should Catholics stand with our Jewish br...	3
874	Los Angeles Dodgers LED Flashlight	0
...	...	...
4946	AntifaTerrorists This hashtag created by corp...	1
10234	It heartening to see in some areas the bush i...	3
1662	Awesome crowd global protest against inequali...	1
9399	How First Australians ancient knowledge can he...	3
966	aux militants de la Je me bats pour vous via ...	4

28338 rows × 2 columns

Figure 10: Data Frame for Participation Levels

Later, I built two separate, but very similar, ELMo models. The first model included four violence classifications and the second model included five participation classifications.

43.2 Embedded Language Modeling

ELMo is a "deep contextualized word representation that models both complex characteristics of word use (e.g., syntax and semantics), and how these uses vary across linguistic contexts (i.e., to model polysemy)"[5]. The word representations in the model are functions of the entire input sentence and are computed on top of a deep biLM with character convolutions, as a linear function of the internal network states[11]. The two-layer construction allows for semi-supervised learning as biLMs are pre-trained on a large text corpus. There are three key features of ELMo [5]:

- *Context* – each word representation depends on the context in which the word is used.
- *Deep* – the representations of words combine all layers of a deep pre-trained neural network.
- *Character-based* – character-based word representations allow the "network to use morphological clues to form robust representations for out-of-vocabulary tokens unseen in training"[5].

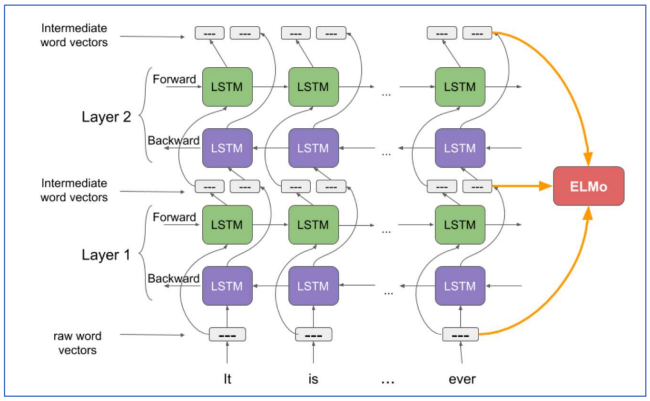


Figure 11: ELMo word vectors computed on top of a two-layer bidirectional language model (biLM)[9]

The pre-trained ELMo model can be found in TensorFlow Hub. The output for an ELMo vector is a 3 dimensional tensor of shape, TensorShape([Dimension(a), Dimension(b), Dimension(1024)]. The first dimension (a) of the output represents the number of training samples. The second dimension (b) represents the length of the maximum string. The last dimension is equal to the length of the ELMo vector. Below are model summaries for both models.

```
Model: "model_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_4 (InputLayer)         [(None, 1)]               0
_________________________________________________________________
lambda_3 (Lambda)            (None, 1024)              0
_________________________________________________________________
dense_6 (Dense)              (None, 256)               262400
_________________________________________________________________
dense_7 (Dense)              (None, 4)                 1028
=================================================================
Total params: 263,428
Trainable params: 263,428
Non-trainable params: 0
```

Figure 12: Model Summary for Violence Levels

```
Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 1)]               0
_________________________________________________________________
lambda_1 (Lambda)            (None, 1024)              0
_________________________________________________________________
dense_2 (Dense)              (None, 256)               262400
_________________________________________________________________
dense_3 (Dense)              (None, 5)                 1285
=================================================================
Total params: 263,685
Trainable params: 263,685
Non-trainable params: 0
```

Figure 13: Model Summary for Participation Levels

As violence levels and participation levels do not impact each other, I created two models that were trained, validated, and tested separately. While the model for violence had four classifying categories, the model for participation had five. Both models' features included five epochs and batch sizes of 256. This means that there were five complete passes through each model. Additionally, looking at the training sets of 20,800 tweets, there were 81 batches with 256 samples and one batch with 64 samples, which means each model was updated 82 times within each epoch. This was repeated four more times to constitute five epochs. Each of the models were validated on 5,200 tweets and tested on 2,388 tweets.

44. RESULTS AND DISCUSSION

With every epoch, training and validation, accuracies increased whereas training and validation losses decreased. Below are the representations of training and validation accuracies, as well as training and validation losses.

44.1 Levels of Violence Model:

Epoch	Training Acc	Training Loss	Validation Acc	Validation Loss
1	0.8728	0.6937	0.9537	0.4394
2	0.9700	0.3424	0.9783	0.2785
3	0.9829	0.2319	0.9831	0.2015
4	0.9874	0.1729	0.9846	0.1583
5	0.9891	0.1415	0.9850	0.1372

Between the first and the fifth epoch, the training accuracy increased by approximately 13.32%, and the validation accuracy increased by approximately 3.28%. The final accuracy of the testing set was 0.9859, which can be interpreted as the model accurately predicted the level of violence for 98.59% of tweets.

If the model included only one epoch, the accuracy of the training data would be 0.8670, and 0.9460 for the validation data. The final test accuracy would be 0.9491. The accuracy of the test data increased by approximately 3.88% when using five epochs instead of one.

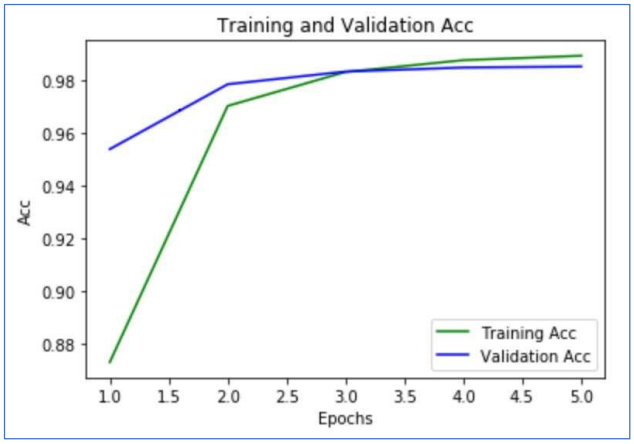


Figure 5: Training and Validation Accuracies for Levels of Violence

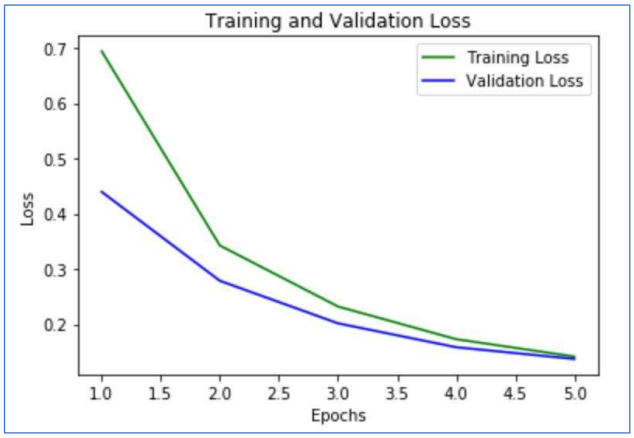


Figure 6: Training and Validation Losses for Levels of Violence

44.2 Levels of Participation Model

Epoch	Training Acc	Training Loss	Validation Acc	Validation Loss
1	0.8530	0.7874	0.9571	0.4679
2	0.9757	0.3662	0.9790	0.2913
3	0.9878	0.2430	0.9871	0.2076
4	0.9913	0.1788	0.9881	0.1640
5	0.9918	0.1461	0.9879	0.1461

Between the first and the fifth epoch, the training accuracy increased by approximately 16.27%, and the validation accuracy increased by approximately 3.22%. The final accuracy of the testing set was 0.9846, which can be interpreted as the model accurately predicted the level of violence for 98.46% of tweets.

If the model included only one epoch, the accuracy of the training data would be 0.8497 and 0.9635 for the validation data. The final test accuracy would be 0.9632. The accuracy of the test data increased by approximately 2.22% when using five epochs instead of one.

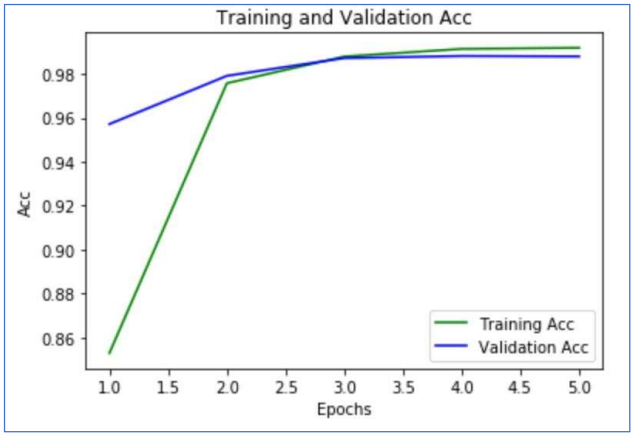


Figure 7: Training and Validation Accuracies for Levels of Participation

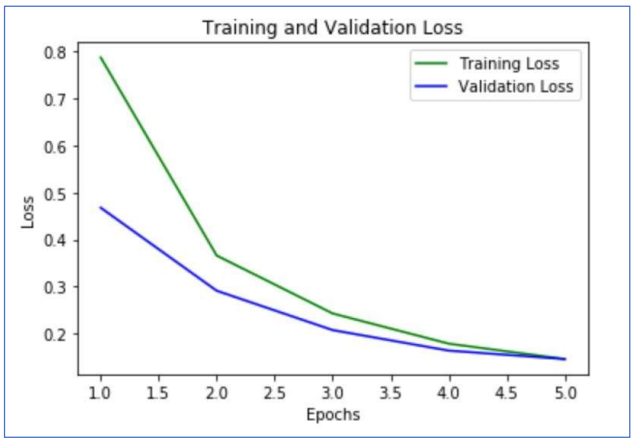


Figure 8: Training and Validation Losses for Levels of Participation

44.3 Both Models

The testing accuracy was higher in the model used for predicting the level of violence. While the difference is insignificant, it may be caused by the different number of classifiers. There are four different levels of violence the tweets were classified into, while there are five levels of participation. When comparing accuracies and losses between using one epoch versus five, we can conclude that five epochs give more accurate results. The model iterates over the data five times, enforcing the training and validation.

45. CONCLUSIONS & FUTURE WORK

ELMo is one of the best performing models in text analysis. It does not only look at traditional word vectors, but also at syntax, semantics, and model polysemy. Linguistic nuances are important to the intelligence community, as the words can have different meaning based on context. The sentence "This movie set was the bomb" has a very different meaning than "I will bomb the movie set." Traditional word embeddings will come up with the same vector for the word "bomb" in both sentences.

Often, it is difficult to go through the data manually. Intelligence analysts do not always have the time or resources to try to pick up on detailed language differences that ELMo can. This research showed that with the use of ELMo, the model can predict the level of violence 98.59% of the time and participation 98.46% of the time. Such information is useful not only to analysts but to entire corporate security departments. Daily, leadership decides whether to keep assets open, suspend work trips, or even evacuate buildings if necessary. If an analyst can determine how violent a demonstration will get based on language changes, the leadership can make appropriate decisions on how to proceed, often impacting the health and safety of the personnel.

There are several improvements to the model that I would like to make in the future. Firstly, I would like to start with k-means clustering to detect trends in tweets in real-time. This would allow for appropriate data collection as the events are happening and not afterward. The clusters that include information on social events such as demonstrations, protests, and rallies would then be used to determine the level of violence and participation. This way, intelligence analysts would be able to witness the escalation of violence and changes in participation in real-time, while observing the events. Additionally, I would like to get more granular as it comes to participation and level of violence and establishes more categories. Currently, there are four levels of violence plus negative data, and five participation categories plus negative data. The smaller the brackets for violence and participation, the more actionable intelligence the analyst has.

Additionally, time constraints and the availability of data with the use of regular Twitter API did not allow for the collection of a larger dataset. Similar language is often used within a social event, therefore, limiting the diversity of words, context, and syntax. The more information collected from different protests and demonstrations, whether based on the reason for the demonstration or country of origin, the broader language spectrum, thereby making the model more comprehensive.

46. REFERENCES

[1] #Charlottesville on Twitter | Kaggle: *https://www.kaggle.com/vincela9/charlottesville-on-twitter#aug15_sample.csv*. Accessed: 2020-04-13.

[2] Aggarwal, C.C. and Subbian, K. 2012. Event detection in social streams. *Proceedings of the 12th SIAM International Conference on Data Mining, SDM 2012*. (2012), 624–635. DOI:https://doi.org/10.1137/1.9781611972825.54.

[3] Bahrami, M. et al. 2018. Twitter Reveals: Using Twitter Analytics to Predict Public Protests. (2018), 1–24.

[4] Baltimore Riots: Social Media and the Crisis on My Doorstep - WSJ: *https://www.wsj.com/articles/baltimore-riots-social-media-and-the-crisis-on-my-doorstep-1430243047*. Accessed: 2019-10-21.

[5] ELMo: Deep contextualized word representations: *https://allennlp.org/elmo*. Accessed: 2020-03-23.

[6] Kimmig, A. et al. 2012. A Short Introduction to Probabilistic Soft Logic. *Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications*. 1 (2012), 1–4.

[7] Korolov, R. et al. 2015. Actions are louder than words in social media. *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2015*. (2015), 292–297. DOI:https://doi.org/10.1145/2808797.2809376.

[8] Korolov, R. et al. 2016. On predicting social unrest using social media. *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social*

Networks Analysis and Mining, ASONAM 2016. (2016), 89–95. DOI:https://doi.org/10.1109/ASONAM.2016.7752218.

[9] Learn ELMo for Extracting Features from Text (using Python): *https://www.analyticsvidhya.com/blog/2019/03/learn-to-use-elmo-to-extract-features-from-text/*. Accessed: 2020-03-23.

[10] Muthiah, S. et al. 2015. Planned protest modeling in news and social media. *Proceedings of the National Conference on Artificial Intelligence*. 5, (2015), 3920–3927.

[11] Peters, M.E. et al. *Deep contextualized word representations*.

[12] van Stekelenburg, J. and Klandermans, B. 2013. The social psychology of protest. *Current Sociology*. 61, 5–6 (2013), 886–905. DOI:https://doi.org/10.1177/0011392113479314.

[13] The Role of Social Media in the Arab Uprisings | Pew Research Center: *https://www.journalism.org/2012/11/28/role-social-media-arab-uprisings/*. Accessed: 2019-10-21.

[14] Won, D. et al. 2017. Protest activity detection and perceived violence estimation from social media images. *MM 2017 - Proceedings of the 2017 ACM Multimedia Conference*. (2017), 786–794. DOI:https://doi.org/10.1145/3123266.3123282.

Voice Cloning for People with Hearing Loss

Minsup Sim
Department of Computing & Information Science
Mercyhurst University, Erie, PA
msim72@lakers.mercyhurst.edu

ABSTRACT

Voice cloning has been a very important topic in artificial intelligence. Among the many possible uses of voice cloning, I propose the use of voice cloning to help people with hearing loss. Through this research, people with hearing loss will be able to copy their voice to other speeches to enable them to talk with others. This research uses MelGAN-VC [6], which adds Siamese network to General Adversarial Network (GAN) to maintain source input's characteristic in an output. This research overcomes the limitation of other speech synthesis methods, which is the source input to the model. This must be spoken by a target speaker. However, a lot of people with hearing loss cannot speak fluently, causing the output of the model to not be a clear speech. The output of this research does not provide perfect speech but rather provides proof of a concept system that can copy the voice of a target speaker to the source input voice.

Keywords

Deep learning, voice cloning, artificial intelligence, generative adversarial network, Siamese network, generator, discriminator

47. INTRODUCTION

According to The Survey of Income and Program Participation (SIPP), there are about ten million hard-of-hearing people and one million functionally deaf in the United States [4]. Only a small portion of them learn to speak clearly. This means that there is a good chance that their families and friends will never hear them talk. A few techniques have been designed to address this problem. This paper focuses on the Text-To-Speech (TTS) technology with voice cloning.

Voice cloning (also called speaker adaptation and speech synthesis and voice fitting) involves making a copy of a speaker's voice using neural networks. Research on voice cloning started recently. In 2017, a faster approach involving neural networks for the whole pipeline was introduced by Baidu Silicon Valley Artificial Intelligence Lab [2]. Compared to prior research that used a mixture of the traditional approach and neural networks, Baidu's research achieved a faster result with little feature engineering. However, the limitation of this approach is that the model requires very long and clear speech as an input to clone the voice. Later Baidu devised with another approach [3], which requires only a few input audios. However, this model also expects at least one full sentence as an input. Since none of the existing voice cloning approaches work for people with hearing loss, a novel approach is introduced in this paper.

The basic idea of the research reported here is to use input voice audio spectrograms to regenerate the target voice's spectrogram using Generative Adversarial Network (GAN). GAN was first used to redraw various images using artistic styles of various famous artists to make it appear as if the images were drawn by those artists. The goal of this work is to use this technique to build a proof of concept system that can copy various linguistic features of the target voice to the source speaker's voice. Enabling people with hearing loss to speak using what source speaker has said.

48. RELATED WORK

Recent research studies have focused on using neural networks to replace the original TTS system. These include WaveNet [7], SampleRNN [8], and Deep Voice [1]. However, this paper focuses on Neural Voice Cloning's speaker adaptation model which relies on a few samples as hard-of-hearing people are only able to generate a small number of inputs.

WaveNet [7] introduced a way of mimicking any human voice and sound generation using a fully convolutional neural network. In the WaveNet model, each convolutional layer has factors to expand receptive field exponentially outputting a prediction for the next audio sequence. The network was trained on English and Mandarin speech waveforms. If another language is used to train the model, the model is designed to produce output in that language.

While WaveNet can get close to human performance, its problem is that it requires a lot of computational resources such as training time and power consumption to achieve this result. To reduce the need for additional resources, a new model called SampleRNN has been developed [9]. SampleRNN consists of recurrent layers, up-sampling layers, and multi-layer perceptrons making it computationally efficient. Those layers are grouped into tiers. In every tier, it receives output from the lower tier to condition the time steps using up-sampling. SampleRNN can not only generate human voices but also instrumental sounds as well.

In 2017, the Deep Voice model [1] was announced by the Baidu research lab. The differences between past models were that Deep Voice does not require a pre-existing TTS system to train a model as well as not requiring hand-engineered features. Deep Voice also solves WaveNet's problem of computationally expensive which allows the model to output predictions in real-time.

The Deep Voice model [1] consists of four different blocks: Grapheme-to-Phoneme block, duration prediction block, fundamental frequency prediction block, and audio synthesis block. When text is given, grapheme-to-phoneme block converts text to phoneme and that phoneme is used as an input for the other three blocks. The output of the duration prediction block and the fundamental frequency prediction block is used as an input with phoneme to be fed into the audio synthesis block outputting audio clip. The audio synthesis block is a modified version of the WaveNet model [7] which fixed WaveNet's problem of taking a long time to generate a sound.

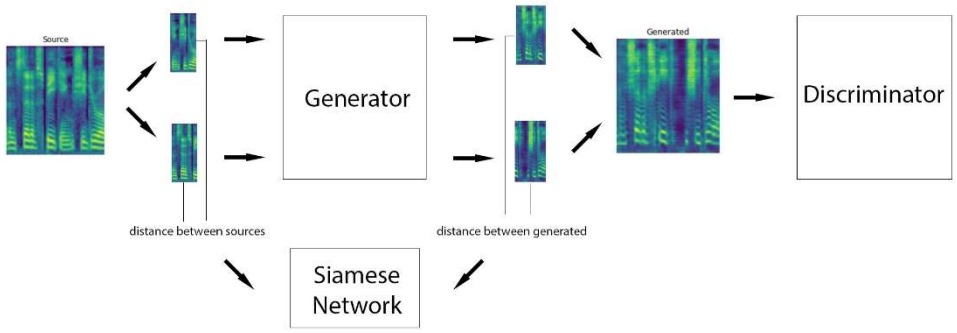


Figure 1: Model diagram. Spectrograms go through three blocks to generate the output

Deep Voice model [1] outputs high-performance sound in fractions of a second. However, the main problem is that hard-of-hearing people cannot speak for hours to create an input. To overcome this problem the approach introduced in this paper is based on another model, MelGAN-VC [6]. This model uses GAN to reconstruct the Mel-spectrogram of source audio file to match the linguistic features of the target voice. The difference between other GANs is that this model has a Siamese network included. Siamese network is originally used for identity classification by finding hidden features of two separate images and minimize the distance of two images' hidden features. This method was often used to prevent GANs from generating the same image as a target since what we want is an image in the style of the target, not the same image as the target. By using the Siamese network, we can minimize the distance between the generated image and the source data to keep features of source data and only apply a certain style of the target to the source data.

One of the ideas to further improve this research was introduced by Baidu in Neural Voice with a Few Samples [2]. This model introduces two approaches: speaker adaptation approach and speaker encoding approach. Speaker adaptation approach uses different accents and dialects to finetune the pretrained multi-speaker model. The speaker embedding part of the model, with few audio-text pairs, is used to adapt to the speaker. While, the speaker encoding approach estimates the speaker embedding of an unseen speaker from scratch. This model uses a regression loss for spectrograms.

Since all models other than Mel-GAN-VC require at least a few sentences to be spoken, it was not possible to use the models as it is. My approach to this problem is to generate few sentences using Mel-GAN-VC after modifying some parts to improve the model, then in the further research, we can feed the output into Baidu's speaker adaptation model to create a voice clone model for people with hearing loss.

49. PROPOSED SOLUTION

The proposed solution that I present in this paper has two phases: First, feature engineering; and second, sentence generation. For evaluation, the results were posted on Amazon Mturk and Survey Monkey to get responses from random people.

49.1 FEATURE ENGINEERING

There are two types of feature engineering tasks needed to carry out this research. One is to convert audio files to Mel-Spectrograms and another one is to divide spectrograms into pieces to feed into the Siamese network. When converting audio files to spectrograms and back to audio files, the method introduced in the inversion of auditory spectrograms, traditional spectrograms, and other envelope representations [3] is used.

49.2 SENTENCE GENERATION

After feature engineering, the input data was fed into GAN model. The generator model generated spectrograms and the discriminator model tried to distinguish if spectrograms generated looks like target spectrograms, while the Siamese model tries to minimize the distance between generated spectrograms and original spectrograms. After the training, the generator outputs an audio file with the voice of a target speaker.

49.3 EVALUATION

Since discriminator is used, I was not able to use the same evaluation method used in Neural Voice with a Few Samples [2] which uses a speaker classification model to classify speakers. The only way to evaluate the output of the generator was to listen to each file. Since the goal of this research is to produce some baseline voice cloning of people with hearing loss, if outputted audio clips contain some similarity in voice by listening to it, which was considered a good result.

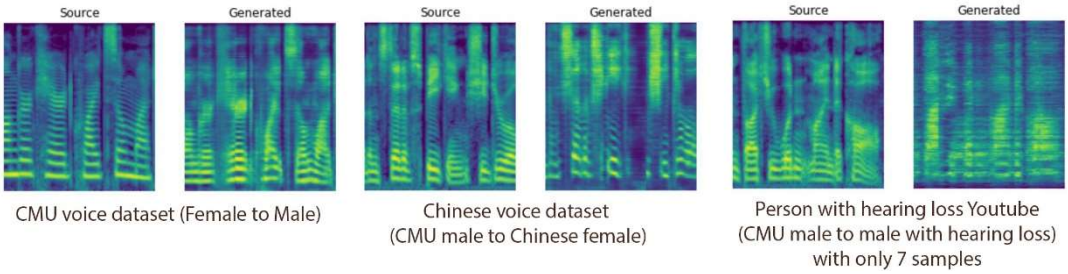
50. MODEL

The whole model includes generator, discriminator, and Siamese network. Generator and discriminator are connected sequentially, and the Siamese network is connected to a generator receiving input of generator and output of the generator as input to calculate the distance between the two. The diagram showing the model is shown in figure 1.

50.1 GENERATOR

The generator model follows the ResNet architecture approach concatenating previous layers to feed into the next layer which is called a residual block. This approach helps to maintain what the model has originally found to be used in deeper layers as well as reducing the vanishing gradient. The overall architecture of the generator is U-Net style. Using residual blocks, model down samples input, and using the convolutional transpose layer, it upsamples back to the original shape.

For convolutional layers, the spectral normalization method introduced in Spectral Normalization For Generative Adversarial Networks [5] is used. This method normalizes weights of convolution layers helping to train the stabilize fast. It is also computationally light making training faster than original convolution layers.



50.2 DISCRIMINATOR

The discriminator is a simple sequential model connecting multiple spectral normalized convolution layers and dense layers at the end to classify if the output of the generator is spoken by the target speaker or not.

50.3 SIAMESE NETWORK

Siamese network receives part of the divided spectrogram as input and trained to find a hidden feature of two source spectrograms to be reduced. In the training GAN model, the output of the generator will be fed into the Siamese network and separate loss function will minimize the distance between two generator output to the distance between two source spectrograms.

Figure 2: source spectrograms and generated spectrograms in different datasets. First two were trained on 1132 samples whereas last one was trained on 7 samples due to lack of resources

51. RESULTS

The model was tested on several voice datasets. Datasets used include:

- Carnegie Mellon University (CMU) dataset,
- single person Chinese voice dataset,
- the voice of a person with hearing loss from Youtube,
- my voice mimicking a person with hearing loss.

The model was able to copy the realistic voice from CMU and single person Chinese voice dataset, but with the voice of a person with hearing loss and voice of myself, the model suffered from generating a realistic result. The difference between datasets that succeeded in generating a realistic voice and that suffered is that the succeeded dataset is recorded in a studio environment with no background noise and echo. Also, the smaller number of samples affected the result as well. In CMU and Chinese voice dataset, 1132 samples of voice were used while in person with hearing loss, only 7 samples were obtained from a single person. The voice data that I recorded contains 400 samples, but because my recordings were not of good enough quality, it did not generate a good result.

The interesting result was that using the Chinese voice dataset, the model showed the possibility to let non-English speakers be able to speak English in their voice.

The source spectrogram and output spectrogram of the generator is shown in figure 2. It shows how close two spectrograms are and how close the generator was able to generate spectrograms with a similar shape with only difference in a voice as well as how a few samples affect spectrogram generation.

Sample outputs were posted on Amazon Mturk and Survey Monkey to collect evaluations on how close and natural the voices are from random people. The results are shown in table 1.

Dataset	Avg Voice Closeness Score	Avg Speech Naturalness Score
CMU	3.11/5.0 *(Mturk)* 3.55/5.0 *(SurveyMonkey)*	2.7/5.0 *(Mturk)* 3.36/5.0 *(SurveyMonkey)*
Chinese	2.7/5.0 *(Mturk)* 4.0/5.0 *(SurveyMonkey)*	2.82/5.0 *(Mturk)* 3.55/5.0 *(SurveyMonkey)*
Person with hearing loss	2.35/5.0 *(Mturk)* 2.72/5.0 *(SurveyMonkey)*	2.47/5.0 *(Mturk)* 2.45/5.0 *(SurveyMonkey)*

Table 5: Amazon Mturk had respondents of 17 and Survey Monkey had 11.

The table shows that CMU and Chinese dataset, which is recorded in studio environment got higher scores on both voice closeness and speech naturalness whereas a person with hearing loss dataset collected from Youtube received lower scores. On speech naturalness, the Chinese dataset received a higher score than CMU dataset and I think it is because CMU dataset's audio clip converted female to male while the Chinese dataset was female to female. I believe converting sex of a voice made the model suffer more than converting in the same sex.

52. DISCUSSION

Since the goal of the research is to design a proof of concept system for voice conversion for people with hearing loss, I believe this project was able to achieve its goal. To improve the result, a new dataset with a clearer recording of the voice of a person with hearing loss is needed.

53. REFERENCES

[1] Sercan Arik, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li,

John Miller, Andrew Ng, Jonathan Raiman, Shubho Sengupta, and Mohammad Shoeybi. 2017. Deep voice: Real-time neural text-to-speech. *34th Int. Conf. Mach. Learn. ICML 2017* 1, Icml (2017), 264–273.

[2] Sercan Arık, Jitong Chen, Kainan Peng, Wei Ping, and Yanqi Zhou. 2018. Neural voice cloning with a few samples. *Adv. Neural Inf. Process. Syst.* 2018-Decem, Nips (2018), 10019–10029.

[3] Rémi Decorsière, Peter L. Søndergaard, Ewen N. MacDonald, and Torsten Dau. 2015. Inversion of auditory spectrograms, traditional spectrograms, and other envelope representations. *IEEE/ACM Trans. Audio Speech Lang. Process.* 23, 1 (2015), 46–56. DOI:https://doi.org/10.1109/TASLP.2014.2367821

[4] Ross E. Mitchell. 2006. How many deaf people are there in the United States? Estimates from the survey of income and program participation. *J. Deaf Stud. Deaf Educ.* 11, 1 (2006), 112–119. DOI:https://doi.org/10.1093/deafed/enj004

[5] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral normalization for generative adversarial networks. *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.* (2018).

[6] Marco Pasini. 2019. MelGAN-VC: Voice Conversion and Audio Style Transfer on arbitrarily long samples using Spectrograms. (2019). Retrieved from http://arxiv.org/abs/1910.03713

[7] Karen Simonyan, Sander Dieleman, Andrew Senior, and Alex Graves. 2016. W n : a g m r a. (2016), 1–15.

[8] Jose Sotelo, Soroush Mehri, Kundan Kumar, Joao Santos, Kyle Kastner, Aaron Courville, and Yoshua Bengio. 2017. Char2Wav: End-to-End Speech Synthesis. *Iclr* October (2017), 44–51. DOI:https://doi.org/10.1227/01.NEU.0000297116.62323.15

[9] Jose Sotelo, Soroush Mehri, Kundan Kumar, Joao Santos, Kyle Kastner, Aaron Courville, and Yoshua Bengio. 2017. Char2Wav: End-to-End Speech Synthesis. *Iclr* October (2017), 44–51. DOI:https://doi.org/10.1227/01.NEU.0000297116.62323.15

About the authors:

Minsup Sim is a data science graduate student at Mercyhurst University. You can find more about him on LinkedIn: https://www.linkedin.com/in/minsup-sim/

Pantograph Pose Estimation

Jesse Decker
Department of Computing & Information Science
Mercyhurst University, Erie, PA
jessedecker@protonmail.com

ABSTRACT

Pantographs provide an electrical conduit from an overhead electric line (OEL) to an electrical motor or battery on vehicles such as locomotives, buses, and trams [1]. They work by dynamically adjusting their position to maintain contact with the OEL and have been used in this manner since electrification in the late nineteenth century.

In this paper, I describe a process for predicting the position of eighteen keypoints on a modern electric railroad pantograph using the Mask R-CNN framework as developed by *He et al*. [2]. Pantographs are an ideal candidate for pose estimation because the components of the pantograph are occluded, or hidden from view, as the pantograph goes through its range of motion. This occlusion makes an exact measurement of the complete object impossible.

Mask R-CNN employs the Microsoft Common Objects in Context (COCO) dataset for object detection, instance segmentation, and person keypoint detection. The framework scored at the top of the results for COCO 2016 in all three tracks of the COCO suite of challenges[3]. Because pantographs are not represented in the COCO dataset, I extend the Mask R-CNN framework to work with a custom dataset.

I trained the network on three hundred images labeled with three classes to represent the points of contact between the pantograph and the OEL. Each class is labeled using bounding boxes, segmentation masks, and a skeleton of six keypoints per class. Pixel distance (PD), or the straight-line distance as measured from labeled keypoint position to predicted keypoint position, was used as a performance metric to evaluate the network. The trained network was able to achieve a mean pixel distance of 16.7 pixels for successful classifications using a validation dataset set of thirty images consisting of 90 detections with 534 labeled keypoints.

Keywords

Mask-RCNN, Pose Estimation, Pantograph, COCO

54. INTRODUCTION

A pantograph is a hinged framework attached to the roof of an electric locomotive used to provide a reliable electrical circuit from overhead electric lines to the electric motor on board the locomotive. The primary components of a pantograph are the panhead that makes contact with the electric wires and two folding mechanical arms that raise and lower the panhead. A lifting control uses inputs from several sensors to operate the arms and maintain contact between the panhead and the OEL.

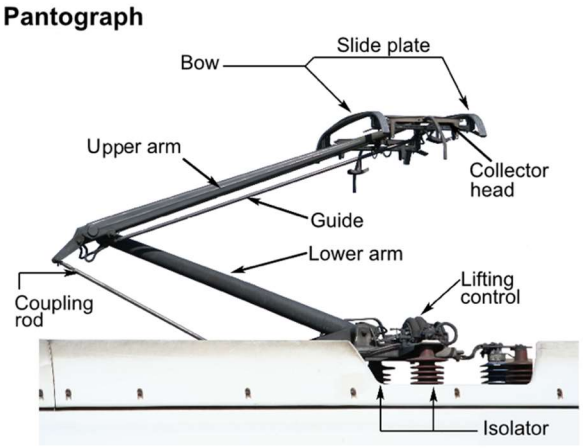


Figure 14: Pantograph diagram

The panhead consists of three components: a front and rear slide plate and a collector head in the middle. The components are fixed and do not move relative to one another.

The panhead is moved vertically to account for changes in distance between the locomotive and the overhead electric wires as it travels down the railway. The panhead can also be moved forward and backward in relation to the locomotive when the lifting mechanism articulates the mechanical arms.

During operation, the panhead can tilt left to right or front to back due to downward pressure from the OEL, vibration, wind, or other misalignments. If the lifting control detects a signal outside of the standard operating threshold, it will collapse the pantograph immediately.

Unfortunately, there are a host of potential hazards in a system which requires a direct physical connection to operate. Disengagement occurs when the panhead loses contact with the OEL. Entanglement occurs when the two become intertwined. Entanglement can be costly in terms of both cost to repair the equipment and for the loss in service time of the impacted locomotive and rail line. A typical contact pattern between the panhead and OEL is visible in Figure 2.

Figure 15: Pantograph sample image

Repair costs are a growing concern as many locomotive manufacturers shift their production from diesel over to hybrid and all-electric engines in a push for efficiency. Newer locomotives also operate at higher speeds, increasing concerns further.

Predicting failure states is a complex problem outside the scope of this paper, but we can better understand the conditions that lead to failure by analyzing how the pantograph moves in space. Pose estimation allows us to do just that by describing the position of several points of interest along the panhead on an X-Y plane. Further, pose estimation provides the ability to output these points even when parts of the panhead are not visible within the frame of the camera.

This data output by the network, a series of keypoints, could be used as input to various computer vision algorithms by railroad engineers if the location of the predicted keypoints is accurate and reliable. To evaluate the ability of the Mask R-CNN network to learn the movement of the pantograph and predict keypoints sufficient for this purpose, I built a framework for labeling images and constructing a dataset, trained the Mask R-CNN network on that dataset, and performed pose estimation analysis using PD as an evaluation metric.

55. RELEVANT WORK

Mask R-CNN is a deep neural-network-based framework for object detection and image segmentation. It extends the object detection framework Faster R-CNN by adding a layer for predicting a segmentation mask from each detection instance [4].

55.1 Faster R-CNN

Faster R-CNN utilizes a two-stage execution to make detections. The first stage, a Region Proposal Network (RPN), is responsible for proposing candidate Region of Interest (RoI) boxes. The second stage utilizes RoI pooling to extract features from these candidate RoIs for classification and regression (Figure 3).

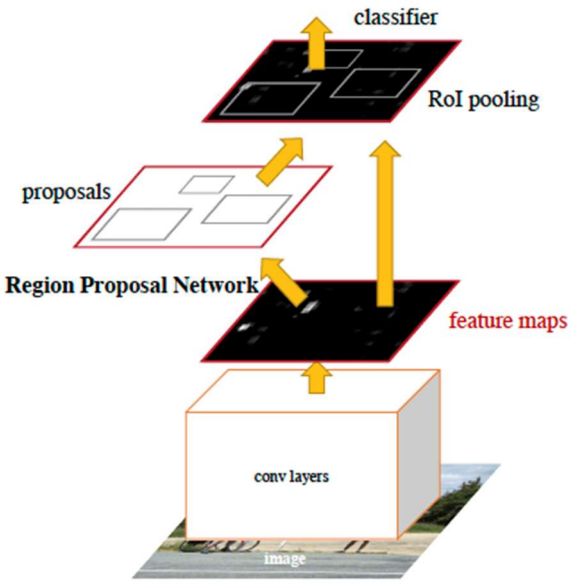


Figure 16: Faster R-CNN[4]

The RPN works by generating a set of rectangular object proposals, each with a regression score, from an input image. A small convolutional is slid across the image at set points, outputting a feature map. The center of each sliding point is called an anchor.

These anchors, along with features data, are used by a classifier to make a series of region predictions to determine the class of object within the region and regression or confidence score. Each region is evaluated by the output score, with the highest scoring classification being selected for prediction.

RoI pooling allows for accurate classification but does not ensure the pixel level alignment between the feature map and RoIs required for image segmentation. For that, we turn to Mask R-CNN.

55.2 Mask R-CNN

Mask R-CNN uses a similar two-stage architecture for detection and segmentation, with the first stage being an RPN similar to the one used in Faster R-CNN. The second stage employs a process called RoIAlign to ensure proper mapping between the extracted features and the input. The significant step forward Mask R-CNN makes over Faster-R-CNN is the addition of a layer parallel to the classification layer that is used to predict a segmentation mask within the RoI (figure 4).

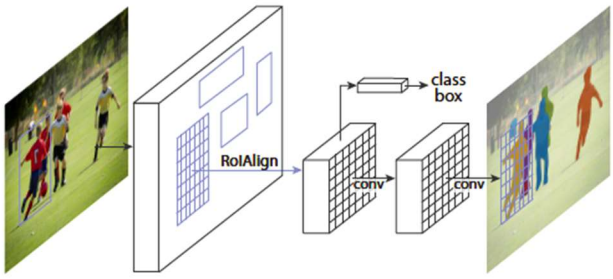


Figure 17: Mask R-CNN [2]

55.2.1 Network Architecture

The network consists of two architectures: the convolutional *backbone* used for feature extraction over an entire image, and the network *head* generating segmentation masks within an RoI.

The backbone relies on a combined ResNet and Feature Pyramid Network (FPN) first stage. The 101 layer ResNet employs five convolutional stages with strides of size 4, 8, 16, 32, and 64. The FPN uses a top-down architecture to build an in-network feature pyramid from a single-scale input.

The head is a set of custom Keras layers that take the RoI as input and outputs a single binary mask for the RoI.

Using a ResNet-FPN backbone for feature extraction with Mask R-CNN provides improved performance with regard to both accuracy and speed in both classification and regression over the previous Fast R-CNN network.

55.2.2 Segmentation Mask

The network adds a branch for predicting a segmentation mask for each RoI in parallel to the classification of the object and bounding box detection regression. The branch is a fully connected layer allowing for pixel-level mapping due to the addition of RoIAlign.

Because predicting the segmentation mask is performed separately from classification, Mask R-CNN does not use the mask to perform classification. This speeds things up to other segmentation models that rely on classification from the mask. Figure 5 shows both the bounding box and the segmentation mask drawn for each detection instance.

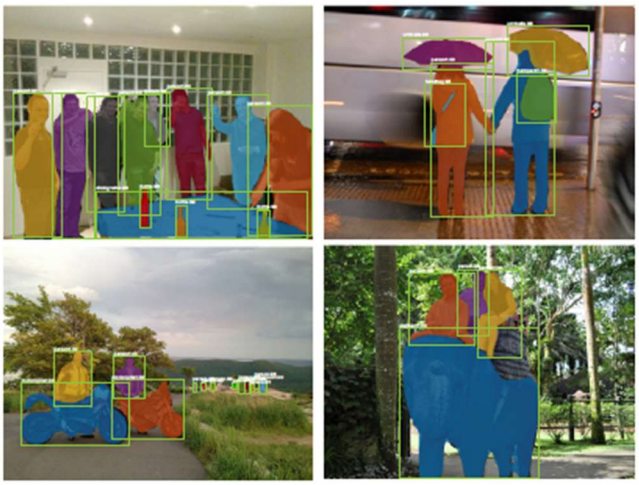


Figure 18: Mask R-CNN classification and detection[2]

55.2.3 Human Pose Estimation

Mask R-CNN extends the functionality of the pixel-level segmentation mask by modeling a keypoint's location as a one-hot mask. Contrary to the segmentation mask itself in which any point can be classified as a foreground class or background object, only a single pixel within the keypoint mask can be classified as a foreground object. The network can predict N number of one-pixel masks within the larger, single segmentation mask as a keypoint for any number of keypoints.

Keypoints are typically used to model human joints such as knee, elbow, and ankle. The network defines a skeleton for visualization as a list of start and end keypoints, i.e., the ankle is connected to the knee. Each human, or person COCO object, is represented by a series of 17 keypoints (Figure 6). The network does not contain any actual domain understanding of the human body, however, and because of this, it can be extended to work with non-person COCO objects.

Figure 19: Mask R-CNN Human Pose Estimation [2]

The keypoint head layers are slightly different from the segmentation mask layer it is based on, utilizing a stack of eight 3x3 512-d convolutional layers, followed by a deconvolving layer to produce a keypoint mask resolution of 56 by 56. This is the same resolution used to store the entire mask. The increased resolution significantly improves keypoint location accuracy.

The network was trained for pose estimation, all 135K images annotated with keypoints in the COCO trainval dataset were used. Images were resized to 800 pixels on the shortest side, and the network was trained for 90K epochs.

	AP^{kp}	AP^{kp}_{50}	AP^{kp}_{75}	AP^{kp}_{M}	AP^{kp}_{L}
CMU-Pose+++ [6]	61.8	84.9	67.5	57.1	68.2
G-RMI [31]†	62.4	84.0	68.5	**59.1**	68.1
Mask R-CNN, keypoint-only	62.7	87.0	68.4	57.4	71.1
Mask R-CNN, keypoint & mask	**63.1**	**87.3**	**68.7**	57.8	**71.4**

Figure 20: Keypoint comparison among top pose estimation models.[2]

Mask R-CNN has shown outstanding results, outperforming other segmentation networks capable of performing pose estimation (figure 7).

56. PROPOSED SOLUTION

My proposed solution for pantograph pose estimation is based on the human pose estimation process as implemented in the Mask R-CNN paper. I developed and iterated through three independent processes for creating a labeled dataset, training the network on the labeled dataset, and evaluating the network's performance using keypoint PD.

56.1 Dataset Creation

The labeled dataset was created using a three-step process of capturing an image from source video, labeling it for image segmentation, and generating annotation data from the labeled image.

56.1.1 Image Selection

All images used to train or validate the network were selected from two video files recorded on a camera located inside an operating

locomotive. It is mounted in a fixed position in the front of a locomotive. The camera is directed toward the pantograph, and the camera angle remains unchanged throughout the videos. As the pantograph travels through its normal range of motion, the panhead moves up and down within the frame of the camera.

The video resolution is 1920x1080 pixels and is recorded in the RGB color mode. Each video is approximately 90 minutes long and is recorded at 60 frames-per-second totaling nearly 650,000 frames.

Each video was recorded in natural sunlight, with no other illumination provided. Image quality is, subjectively, quite good. Three issues that appeared intermittently within the dataset that could impact the performance of the network include low light conditions, background interference with catenary structures, and instances where the panhead is partially out of frame.

I first developed an algorithm to save frames of video as JPEG images. An equal number of randomly selected frames from each of the source videos are saved into train and validation folders at full resolution. The video name and frame number were combined to form a unique file name.

56.1.2 Image Labeling

Because neither the panhead nor its components are defined in the COCO dataset, I defined three classes to represent the components of the panhead: front bar, middle bar, and rear bar. Due to the vertical movement of the panhead, one bar can block, or occlude, the view of another bar with respect to the camera.

Occlusion occurs regularly in the dataset with the front bar occluding the view of some part of the middle bar and the middle bar occluding the view of some part of the rear bar. This is visible in figure 8, with the front bar in red, the middle bar in green, and the rear bar in blue.

Figure 21: Occlusion of the middle and rear bars.

For each instance of a class visible in an image, a series of color-coded annotations are drawn. A single rectangular bounding box is added to classify the instance and define the outer boundaries of the instance. One or more polygonal masks are added to specify the visible portion of the object.

A set of color-coded one pixel by one-pixel keypoints are drawn to define the location and visibility of points of interest on the instance. A keypoint can exist in one of three possible states of visibility: visible within the image, not visible within the image, and occluded within the image. Each state is distinguished with a corresponding color.

All three classes share a skeleton of six labeled keypoints: L1, L2, L3, R1, R2, and R3. Keypoints L1 and R1 represent the outermost edge of the bar in each class. Keypoints L2 and R2 represent a downward arc at the mid-point of the bar visible in each instance. Keypoints L3 and R3 represent the flat, horizontal section of the bar for each class.

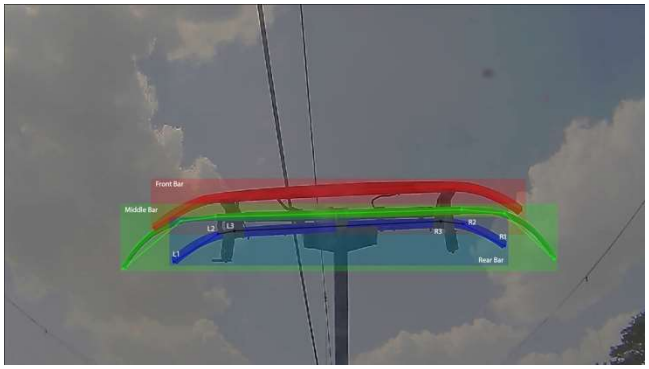


Figure 22: A fully annotated image

The L3 and R3 keypoints of the rear bar are always occluded and, therefore, always estimated even when labeled manually. The position of the L2 and R2 keypoints for all three classes is challenging to determine consistently. The L1 and R1 keypoints are generally easy to identify but are subject to being cropped out of view when the panhead is at the bottom of the camera frame, and therefore labeled as not visible. This may impact results as the dataset is already limited.

All image labeling was done in Adobe Photoshop, and each annotation is added as an image layer. Once labeled, each annotation layer is exported as a bitmap image and saved alongside the original JPEG image. Bitmap images are used because they provide an anti-aliasing free means of defining edges necessary for generating pixel-level data. The fully annotated Photoshop PSD file is saved for future use as needed. Photoshop actions are used to automate much of the process.

56.1.3 Data Generation

I developed an algorithm to read the type and position of annotation encoded in each of the bitmap images. Annotation data is formatted to be consistent with the COCO data format with images, classes, and annotations being saved to JSON file.

Both automated and manual data validation processes were employed to ensure alignment between labeled image and annotation data. Each master image is opened, and all annotations are overlaid onto the image for a visual inspection. An algorithm ensures all keypoints fall within their respective segmentation mask and that image and annotation data remain in alignment as updates are made to the labeled images. After inspection and validation, all bitmap images associated with a master image are deleted to save file space and must be exported from the PSD file if needed.

The completed dataset consists of 330 images split into a training set of 300 images and a validation set of 30 images. The training set consists of 900 labeled detections with 5290 keypoints, and the validation set has 90 labeled detections with 534 keypoints.

56.2 Network Development

Before working with the Mask R-CNN network, several changes were required to the underlying codebase to allow for the training of the three custom classes and skeleton.

A pantograph class was extended from the base COCO class, and the data loading process was modified to work with the default COCO format. During this data loading process, the three labeled

classes are read from the annotation file into the PantographDataset class. The list of keypoints and the list of skeleton connections are read in as class attributes.

A PantographConfig class was written to override the default configuration settings as needed. The number of classes was specified as 4, three custom classes plus the background class required for segmentation. The number of keypoints was specified as 6. The maximum dimension image setting was set to 1024.

Several changes were required to the Mask R-CNN model, utility, and visualization files as well. Throughout the files, hard-coded references were updated to correspond to the appropriate configuration variable. Several functions responsible for resizing the image, mask, and keypoints were updated to work with more modern packages. All hard-coded references to the keypoints and skeleton were updated. I incorporated my own data visualization code into the default code to allow for more control over the process and further assure the alignment of the data as it moves from data creation to model development.

56.2.1 Configuration

I employed a data inspection process to determine optimal configuration settings before training. With a small sampling of images chosen at random from both the train and validation datasets, I tested the updated PantographDataset class methods and Mask R-CNN code in the order they are run when training. The images and any accompanying annotations and data are printed to screen for visual inspection.

Each image is resized to 1024 by 1024 pixels with padding added to the top and bottom of the image to avoid cropping any part of the image. Resizing the images was required to fit multiple images in a batch using a GPU. During this process, resized bounding boxes, segmentation masks, and keypoints are computed.

The use of mini-masks further reduces the amount of memory needed by storing the information about the segmentation mask in an N by N mini mask instead of a full-sized 1024x1024 mask. This offers substantial memory savings since there are generally three positive class instance for each image. Some loss of fidelity can occur during this resizing process, mainly when dealing with smaller masks, as is the case with the middle and rear bars. This loss of fidelity could translate to a reduction in the number of training points available to the network. This loss is also noticeable when the mini-mask is resized to 1024 by 1024 pixels and displayed on the screen. I settled on a mini mask shape of 224 by 224 pixels as a compromise between fidelity and the amount of memory available.

Mask R-CNN supports one generator-based data augmentation technique by default, horizontal flip. In testing, I found this technique added variation to the training information that never occurs in this pantograph dataset. Horizontal flip was disabled before training. I considered other techniques, including equalizing the histogram of the image, but concluded this dataset would not make a good candidate for augmentation due to the limited, but particular variation within the dataset. Training the network on this dataset would benefit from a higher number of images labeled training images rather than data augmentation.

56.2.2 Training

The network was initialized with the weights file provided by the paper's author and was trained using the Google Cloud Platform with a single Nvidia Tesla V10 GPU. With a batch size of 2 images, I was able to process all of the images in each epoch in 150 steps per epoch, training both the backbone and head for a total of roughly 400 epochs. This took approximately 10 hours spread over several sessions. After training both the Resnet backbone and custom head layers, I froze the backbone and trained the head for another 50 epochs.

The final training loss was 1.6, and the validation loss was 1.8.

56.3 Network Evaluation

The trained network was used to make predictions on the labeled validation dataset. The resulting dataset, including predicted classes, bounding boxes, masks, and keypoints results for each of the 30 validation images, was saved to JSON file, then loaded into an evaluation script alongside the labeled JSON file. Pose estimation analysis was performed to compare the performance of the network using the two files.

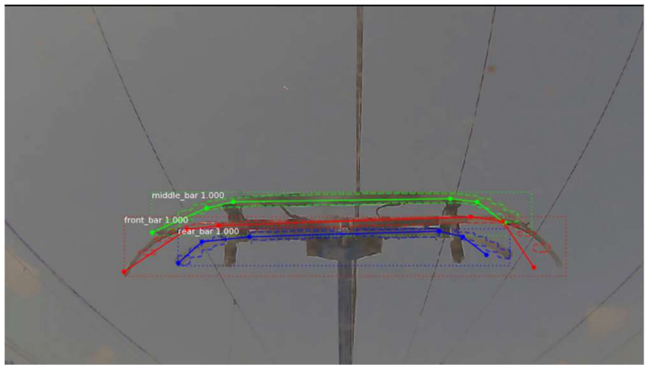


Figure 23: Detection showing misclassification of front and rear bars with segmentation masks and keypoints.

Because the classification and regression score are generated by the network's backbone, and the segmentation mask and keypoints are generated by the head layers, we can evaluate the network's ability to classify instances and its ability to ability to generate masks within the instance separately.

56.3.1 Classification

The first step in evaluation is comparing the labeled classifications to the predicted ones. Comparing figures 10 and 11, we see the network only made 83 detections, seven short of the labeled 90. This is due in part to maintaining a minimum confidence hyperparameter of .9 for detection. Lowering this threshold increased the number of detections, but at the cost of accuracy. At .9, this makes for a successful classification rate of .59.

```
Validation Dataset
------------------
Number of images:        30
Number of detections:    90
Number of keypoints:     534
```

Figure 24: Validation dataset

```
Predicted Dataset
-----------------
Number of images:        30
Number of detections:    83
Number of keypoints:     498
```

Figure 25: Predicted dataset

56.3.2 Pose Estimation Analysis

The second step is to evaluate the network's ability to predict keypoints within a segmentation mask. To do this, I used only the keypoints associated with successful classifications. As before, a hyperparameter can be adjusted to optimize the balance between the number of keypoints predicted and their accuracy. In this subset of data, there are 291 keypoints.

```
count    291.000000
mean      16.748488
std       13.112395
min        0.000000
25%        5.915000
50%       15.030000
75%       21.010000
max       52.090000
Name: PD, dtype: float64
```

Figure 26: Pixel distance summary statistics

Figure 12 shows the mean pixel distance for the predicted points was 16.7 pixels, with 50% of the keypoints being within 15 pixels. To provide a frame of reference for evaluation with regard to solving the original problem, we need to understand what is a usable distance and how many keypoints fall within that distance. The maximum usable pixel distance to build pantograph failure detection systems is 20 pixels. A distance of 5 pixels or less is required to confidently build more accurate computer vision algorithms, including 3D scene reconstruction.

```
PD<=      Count   PCGT
20        210     : 72.16494845360825
10        103     : 35.39518900343643
5         60      : 20.618556701030926
1         5       : 1.718213058419244
```

Figure 27: Keypoints by pixel distance

Figure 14 shows that 72% of the keypoints met the 20-pixel requirement, while just over 20% of the keypoints met the 5-pixel requirement. Five of the predicted keypoints were at 1 pixel or less away from their labeled position.

To further analyze the network and understand where improvements can be made, I broke the results down by class and keypoint.

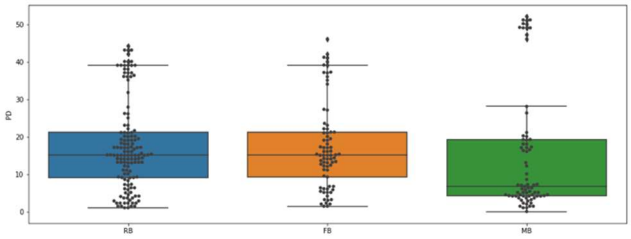


Figure 28: Pixel distance by class

Figures 15 shows a boxplot with values overlaid. From this chart, we can see the rear and front bars on the left and center performing similarly. Both have a roughly even distribution of pixel distances with a close mean as well. The middle bar, however, shows much a much lower mean of 9 but is unevenly distributed. This class has many keypoints at a pixel distance of 5 or less, but also more than ten outlier keypoints greater with a pixel distance greater than 45.

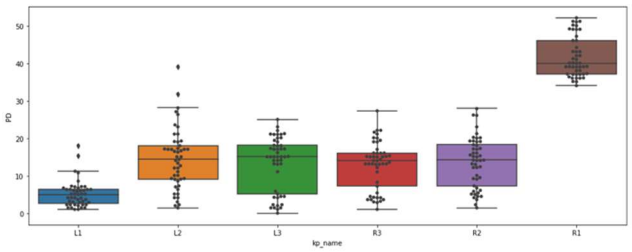


Figure 29: Pixel distance by keypoint

Figure 16 shows a breakdown of pixel distance by keypoint. From this chart, we can see a shared range of values across the middle sections of the bars and a stark difference in values for the left and right most keypoints. This is confirmed by visual inspection of a sample image and its keypoints.

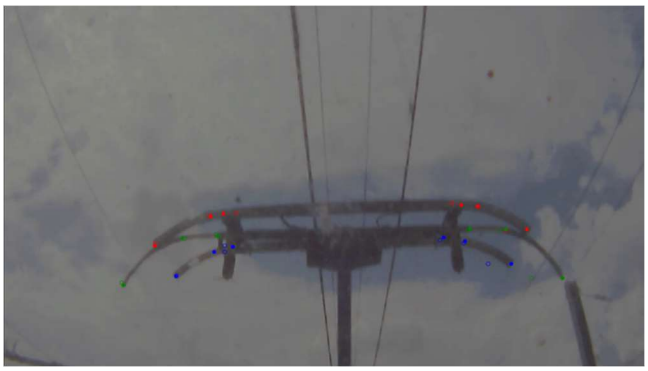


Figure 30: Labeled and predicted keypoints

Figure 17 shows a validation image and its labeled keypoints represented as solid circles and predicted keypoints as stroked circles. Here, we see the increase in pixel distance for each keypoint from left to right.

57. CONCLUSIONS

I have demonstrated the ability of the Mask R-CNN network to learn the movement of the three pantograph components and to successfully predict the position of a series of keypoints associated with each class. While not sufficient for production-level work, the results prove the network is capable of the task. The classification results validate the need for additional labeled data to train the Resnet backbone. Pose estimation analysis shows that a high degree of accuracy is possible given accurate region data from the backbone.

58. FUTURE WORK

I believe the network would show improved results with increased training data. I would recommend a continued iterative approach to training and evaluation until reaching a satisfactory pixel distance measurement.

My code-to-image editor-to-code solution for annotation has proven to be a reliable alternative to commercial and open source options. It could easily be adapted to annotate other custom classes using a variety of image editing tools.

I would discuss changing the position and alignment of the camera with the owner. Due to the distance from the camera, the smaller parts of the panhead bars are proving challenging to detect. Additionally, I would advise correcting the alignment between the camera and pantograph to reduce the offset distances between the left and right-hand sides of the pantograph.

59. BIBLIOGRAPHY

[1] D. He, Q. Gao, and W. Zhong, "A Numerical Method Based on the Parametric Variational Principle for Simulating the Dynamic Behavior of the Pantograph-Catenary System," *Shock and Vibration*, 2018. https://www.hindawi.com/journals/sv/2018/7208045/ (accessed May 07, 2020).

[2] "[1703.06870] Mask R-CNN." https://arxiv.org/abs/1703.06870 (accessed Mar. 04, 2020).

[3] T.-Y. Lin *et al.*, "Microsoft COCO: Common Objects in Context," *arXiv:1405.0312 [cs]*, Feb. 2015, Accessed: Mar. 09, 2020. [Online]. Available: http://arxiv.org/abs/1405.0312.

[4] "[1506.01497] Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." https://arxiv.org/abs/1506.01497 (accessed Mar. 05, 2020).

About the authors:

Jesse Decker is a graduate student at Mercyhurst University.

Machine Learning in Money Laundering Detection

Huyen Vu
Department of Computing & Information Science
Mercyhurst University
hvu15@lakers.mercyhurst.edu

ABSTRACT

60. Money laundering creates serious threats to both financial institutions and nation-states. Hundreds of billions of dollars of criminal money have been laundered through financial institutions annually. To prevent money laundering activities, developing an effective technique to detect suspicious transactions is crucial for financial institutions. Despite the resources and efforts spent by financial institutions, many cases of money laundering are still left undetected. Therefore, preventing and detecting suspicious activities is one of the most significant challenges for financial institutions. This paper presents the application of machine learning in money laundering detection by classifying financial transactions into two categories: fraudulent and non-fraudulent. The models used in this paper are deep learning, Random Forest Classifier, Logistic Regression, and Support Vector Machines. The deep learning model and Random Forest Classifier model have shown a high accuracy number with promising results in reducing the number of false positives.

Keywords

Anti-money laundering, money laundering detection, machine learning, deep learning, classification, fraud detection, cryptocurrency

61. INTRODUCTION

Money laundering is the process of making illegal money appear legal, a method that is used by criminals and terrorists around the world. Illegal money comes from criminal activities, including drug trafficking, terrorist activities, illegal tax avoidance, and any other crimes [1]. Criminals with income from these illicit sources try to integrate their "dirty" money into the legitimate financial system. General fraud, narcotics trafficking, embezzlement, and cybercrimes are the most common crimes related to money laundering [2]. Genzman defined money laundering as an activity that involves knowingly engaging "in a financial transaction with the proceeds of some unlawful activity with the intent of promoting or carrying on that unlawful activity or to conceal or disguise the nature location, source, ownership, or control of these proceeds" [3].

Money laundering creates serious threats not only to financial institutions but also to nation-states. Financial institutions could be exposed to various types of severe risks, such as operational risk and legal risk. "Money laundering can erode the integrity of a nation's financial institutions and can also adversely affect currencies and interest rates" [4]. At the societal level, the money that comes from the act of money laundering can be used to fund criminals to operate and expand their criminal networks. According to the statistics on financial crime by PwC, this includes:

- $150B in annual profits (USD) generated from forced labor as a result of human trafficking,
- $2B the peak estimated size (USD) of the global illicit drug market in 2017,
- $6.6B the estimated money (USD) generated annually by smugglers along Latin America's routes in the U.S.,
- $652B the estimated size (USD) of the global illicit drug market in 2017,
- $1.26T the estimated annual cost (USD) of corruption to developing countries [5].

The United Nations Office on Drugs and Crime (UNODC) conducted a study to determine the number of illicit funds generated by drug trafficking and organized crimes and to investigate the amount of funds are laundered [6]. The report estimates that in 2009 alone, criminal proceeds amounted to 3.6% of global GDP, with 2.7% (equivalent to USD 1.6 trillion) being laundered [7]. It cannot be denied that money laundering creates economic instability for nation-states. The economic effects of money laundering include undermining the legitimacy of the private sector, undermining the integrity of financial markets, loss of control of economic policy, economic distortion and instability, loss of revenue, risks to privatization efforts, and reputation risk [4]. Therefore, the prevention of money laundering is crucial in order to protect national financial stability and international security. "Preventing the financial system from being misused to launder illicit funds or to fund terrorist attacks is a key feature in the global effort to reduce the devastating effects of crime and terrorism" [8]. Due to the high amount of transactions and the variety of money laundering techniques, the number of undetected illicit funds still remains too high, and it is difficult for the authorities to manually detect money laundering activities and prosecute the criminals.

In general, money laundering is broken down into three stages: placement, layering, and integration [9]. First, "placement" is the process of entering illegal money into the financial system. This is the riskiest phase since it shows a direct connection to the money source. Second, "layering" is the process of separating the money from its illegal source through a series of financial transactions. This is the most complex phase since the origin of the money is

made difficult to trace back. And third, "integration" is the illegal funds coming back to the criminals, looking like they are from legitimate sources. Because laundering money almost always requires it to pass through one or more banks, the primary strategy to fight against it is to require banks to monitor transactions to make sure their accounts are not being used for money laundering. In cases of high-risk transactions, banks may be required to file a suspicious activity report (SAR) with law enforcement [9].

Besides laundering money through financial institutions, criminals have chosen cryptocurrencies as a popular option to launder money due to its growth in prominence and price. Data shows that criminals have laundered over $2.5 billion worth of dirty Bitcoins through the unregulated cryptocurrency exchanges [10]. "Many criminals used Bitcoin's pseudonymity to hide in plain sight, conducting ransomware attacks and operating dark marketplaces for the exchange of illegal goods and services" [11]. There are many cryptocurrency intelligence companies focused on protecting the cryptocurrency ecosystems from criminal activities. Researchers have conducted a number of research studies to increase the effectiveness and accuracy of classifying Bitcoin transactions.

The Bank Secrecy Act (BSA) requires financial institutions to detect and report customers engaged in money laundering, fraud, terrorist financing, and sanction violations. Traditional money laundering detection or anti-money laundering (AML) systems are rule-based systems that create alerts and flag transactions as suspicious using pre-determined rules [12]. Those flagged transactions are manually checked by an investigator to determine whether a SAR report needs to be filed. A specialist investigates the alerts to determine if an alert is a genuine money laundering case, or whether it is a false positive. This makes a traditional AML system is heavily labor-intensive. Since banking transaction data comes in an enormous volume, human analysts need automated tools to detect money laundering patterns more accurately and effectively. "The issue with relying on a rule-based system to tackle money laundering is these traditional systems are rigid and cannot adapt to continuously changing data" [12]. Having a high number of false positives is time-consuming and costly for financial institutions. Therefore, financial institutions want a way to reduce the number of false positives while still following the rules, so that any potential cases of money laundering are not left undetected. In this case, data science can help improve AML solutions.

There are numerous benefits of using data science in an AML system: reducing the number of false positives to concentrate manual investigations on high-risk alerts, combining available data sources to create a better picture of the overall transactions, and flexibility and adaptability of machine learning models as compared to traditional models [12]. Financial institutions are increasingly turning to advanced analytics and machine learning to help them fight money laundering. "Machine learning techniques hold great promise in addressing some of the challenges financial institutions are grappling with. They can be used to increase the efficiency of measures in the various elements of the AML framework, for example, to reduce false positives and improve the effectiveness of transaction monitoring" [8]. Banks can use machine learning to monitor transactions for every account in real-time, thus preventing the crime from happening. Through the use of machine learning, a system can be trained to detect a large number of micropayments that are suspected to be involved in money laundering activities.

To help financial institutions detect money laundering patterns in large volumes of data, as well as to adapt to changes in criminal activities, this paper aims to analyze the use of machine learning techniques in money laundering detection by applying machine learning models on sample financial datasets. This paper is divided into five main sections. Section 1 introduces money laundering and the problems caused by money laundering activities. Section 2 provides previously presented research in the area of money laundering detection. The proposed approach to analyze financial transactions is presented in Section 3, and the results of the proposed approach are discussed in Section 4. Finally, Section 5 concludes the paper and provide ideas for future research.

62. RELEVANT WORK

Given the importance of preventing money laundering activities and the complexity of effectively identifying money laundering patterns, it is essential to identify various methods that help the process of anti-money laundering. In this section, an overview of previously presented research in the area of money laundering detection will be presented. A variety of machine learning approaches to aid anti-money laundering efforts have been introduced and published in the past years. However, building an efficient anti-money laundering system remains a significant challenge for financial institutions due to a huge volume of transactions and changes in patterns of criminal activities.

Rule-based methods: Rule-based methods include both classification and prediction methods [13]. One of the earliest studies in the field of money laundering detection was presented by Senator *et al.* [14]. In this research, the authors proposed FinCEN (Financial Crimes Enforcement Network) - an artificial intelligence-based system that applied rule-based Bayesian model evaluation to identify suspicious transactions from reports of large cash transactions. Panigrahi *et al.* [15] proposed a system for database intrusion detection using a combination of rule-based approach, belief, historical database, and Bayesian learning to collect customers' transaction behavior and mark suspicious transactions. Later, Khan *et al.* [16] presented a Bayesian approach for suspicious financial activity reporting. In this study, the authors created a model based on customers' transaction history. This model was used to predict future activities of current and future customers. If any future transactions have any patterns that are significantly different from historical transactions, those transactions are marked as suspicious. In the study by Rajput *et al.* [17], the authors proposed an ontology-based system that uses semantic web technologies and domain knowledge for detecting suspicious transactions by monitoring independent transactions.

Classification-based methods: Classification methods such as decision trees [18, 19, 20], support vector machine [21, 22], and neural networks [23] have been used to identify money laundering patterns. Wang & Yang [18] determined if the transaction was considered money laundering activity by using decision tree in their study. Lopez-Rojas and Axelsson [19] applied decision tree learning and clustering techniques to analyze synthetic mobile money transaction datasets. Liu *et al.* [20] used the combination of decision trees along with K-means and BIRCH clustering algorithms to determine money laundering transactions. Support vector machines is a supervised learning method used for classification [13]. Tang & Yin [21] and Keyan & Tingting [22] presented a support vector machines-based method to detect

suspicious activities. The neural network methods use a set of connected nodes to learn a function. Lv *et al.* [23] presented an approach using neural networks with the method of Radial Basis Function (RBF), including three layers to detect money laundering activities. However, the limitation of these approaches is that these systems can only detect patterns similar to those that have been observed in the past.

Clustering-based methods: Clustering methods classify data into different groups such that data in the same group have the most similarity to each other, and the data in different groups have the least similarity to another group [13]. In AML, clustering is used to group transactions and/or accounts into clusters based on the similarities they share. This technique helps to detect patterns of a group of suspicious transactions. Zhu [24] created a profile for customers, then compared every customer transaction with the same customer's transaction history to detect money laundering activities. Cao & Do [25] applied clustering financial data using the CLOPE algorithm to detect money laundering.

Inspired by the previous research on anti-money laundering, a detailed proposal of a model that can be used to analyze financial transactions data and detect suspicious activities will be discussed in the next section.

63. THE APPROACH

This section presents the approach to analyze financial transactions to detect fraudulent or non-fraudulent transactions that is adopted in this paper. In order to study the effectiveness of the approach, two different datasets will be conducted. The approach consists of three parts: data preprocessing, model building, and model evaluation. Different machine learning algorithms will be used to classify a fraudulent transaction and a non-fraudulent in this research are deep learning Keras, Logistic Regression, Support Vector Machines, and Random Forest Classifier. The exploratory data analysis will also be conducted to get better insights of the dataset. The two datasets that are used to analyze in this paper are the "Synthetic Financial Datasets for Fraud Detection" [26] and the "Elliptic Data Set" [27].

63.1 Experiment 1 – Synthetic Financial Datasets for Fraud Detection

Experiment 1 was conducted using the "Synthetic Financial Datasets for Fraud Detection." While it would have been preferable to use a real-world data set, no such data set is available due to data privacy and data ownership issues. The synthetic dataset consists of several features, including:

- step: 1 step is 1 hour of time,
- type of transaction: CASH_IN, CASH_OUT, DEBIT, PAYMENT, TRANSFER,
- amount: the amount of the transaction in local currency,
- nameOrg: customer who starts the transaction,
- oldbalanceOrg, newbalanceOrg: the account balance before and after the transaction,
- nameDest: recipient of the transaction,
- oldbalanceDest, newbalanceDest: the account balance before and after the transaction,
- a feature that classifies a fraudulent transaction as 1 and a non-fraudulent one as 0 to train the algorithm.

Since the dataset is quite large, a sample of 100,000 randomly chosen instances was used to work with.

63.2 Experiment 2 – Elliptic Data Set

Experiment 2 studied the effectiveness of various machine learning approaches on the Elliptic Data Set. It is a public Bitcoin transaction graph dataset with real entities, and the task for the dataset is to classify illicit and licit nodes in the graph. The dataset maps Bitcoin transactions with 203,769 nodes and 234,355 edges [27]. The nodes are categorized into three classes, including "illicit", "licit", and "unknown" [27]. According to the description of the dataset, two percent (4,545) of the nodes are labeled "illicit", twenty-one percent (42,019) of the nodes are labeled "licit", and the remaining transactions are "unknown" [27]. There are 166 features associated with each node [27]. The exact description of the features is not provided due to intellectual property issues. The first 94 features represent local information about the transaction including the time step, the number of inputs/outputs, transaction fee, output volume and aggregated figures, and the average number of incoming (outgoing) transactions associated with the inputs/outputs [27]. The remaining 72 features are aggregated features [27].

63.3 Data Preprocessing

For data preprocessing, each dataset was first loaded into a pandas data frame, and then any missing values from the datasets were checked and removed. Exploratory data analysis was then performed. For the next step, dummy variables for categorical values were created. Then, the dataset was split into training set and test set with 80% for training and 20% for testing. After splitting the dataset, StandardScaler from sklearn.preprocessing was applied to normalize the data before training for the deep learning Keras model.

63.4 Deep Learning Keras Model

For the deep learning Keras model building, the model was created by applying "relu" and "sigmoid" as activation functions since they are considered to be more effective when working with binary classification problems, and using "adam" for the optimizer, "binary_crossentropy" for the loss function, and "accuracy" for metrics. Other classification algorithms, including Logistic Regression, Support Vector Machines, and Random Forest Classifier were also applied to the training set to compare how the deep learning Keras algorithm performed.

63.5 Model Evaluation

To evaluate performance, the accuracy scores were examined. To evaluate the importance of features on the classification task, "feature_importances_" was computed using the scikit-learn library from Random Forest Classifier model.

64. RESULTS & DISCUSSION

This section discusses the findings and results of the two experiments conducted in this research.

64.1 Experiment 1

After training the training set on the deep learning Keras model, the accuracy of the model on the test set is 99.92%. The Random Forest Classifier also gives a high accuracy percentage of 99.95%. The accuracy scores for Logistic Regression and Support Vector Machines are 98.59% and 99.78%, respectively. Table 1. presents the results of experiment 1.

Model	Accuracy
Deep Learning Keras	99.92%
Logistic Regression	98.59%
Support Vector Machines	99.78%
Random Forest Classifier	99.95%

Table 1. Results of Experiment 1

After running the "feature_importances_" on the Random Forest Classifier model, the "newbalanceDest" and "oldbalanceOrg" features have the most influence and the type of transaction "PAYMENT" and "DEBIT" features have the least influence. Table 2. shows the importance of the features from the most important to the least important.

Feature	Importance
newbalanceDest	0.281
oldbalanceOrg	0.249
amount	0.152
step	0.128
oldbalanceDest	0.073
transfer	0.046
newbalanceOrg	0.039
cash_out	0.023
cash_in	0.006
payment	0.002
debit	0.000

Table 2. Importance of features

The importance of each feature on the dataset can help the human investigator know which features they should pay more attention to when detecting money laundering cases.

After performing the exploratory data analysis on the dataset, some interesting results were founded. Types of fraudulent transactions are "TRANSFER" and "CASH_OUT" with 4,097 and 4,116 fraud cases, respectively. The minimum amount "isFlaggedFraud" is 353874.22.

64.2 Experiment 2

For this experiment, the deep learning Keras model also gave a high accuracy score of 97.01%. After running Logistic Regression, Support Vector Machines, and Random Forest Classifier on the dataset, the Random Forest Classifier model gave the highest accuracy score of 97.53%. The accuracy score of Logistic Regression and Support Vector Machines are 95.39% and 96.96%, respectively. Table 3. presents the results of Experiment 2.

Model	Accuracy
Deep Learning Keras	97.01%
Logistic Regression	95.39%
Support Vector Machines	96.96%
Random Forest Classifier	97.53%

Table 3. Results of Experiment 2

In this experiment, since the specific label for each feature is not disclosed, it is difficult to determine which feature is the most important and the least important. Also, there are 166 features in this dataset, so the importance of each feature is relatively small. Table 4. shows the 10 most important features in this dataset.

Feature	Importance
tx_feat_48	0.053
tx_feat_56	0.044
tx_feat_19	0.043
tx_feat_54	0.043
tx_feat_50	0.042
tx_feat_6	0.035
tx_feat_77	0.034
tx_feat_91	0.028
tx_feat_44	0.024
agg_feat_45	0.023

Table 4. 10 most important features

By looking at Table 2., the features represent local information about the transaction are mostly the most important features. Investigators should pay more attention to these features when investigating fraud.

64.3 Keras Model

The figures below show the model accuracy and model loss for the deep learning Keras model. It can be seen that the model was learning at each and every epoch and minimizing the loss.

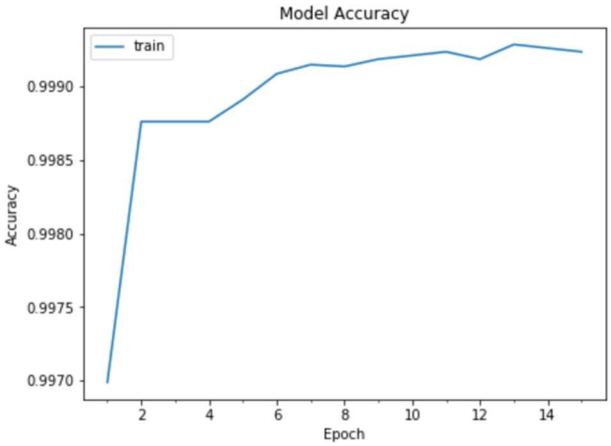


Figure 1. Keras model accuracy for Experiment 1

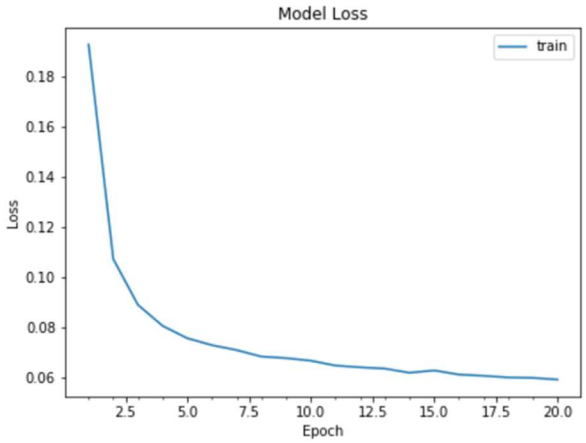


Figure 2. Keras model loss for Experiment 1

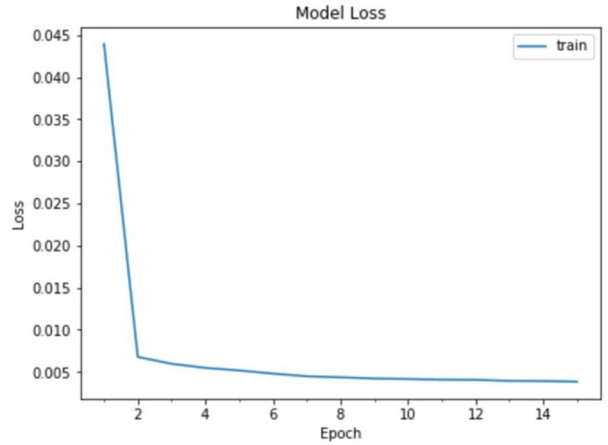


Figure 2. Keras model loss for Experiment 1

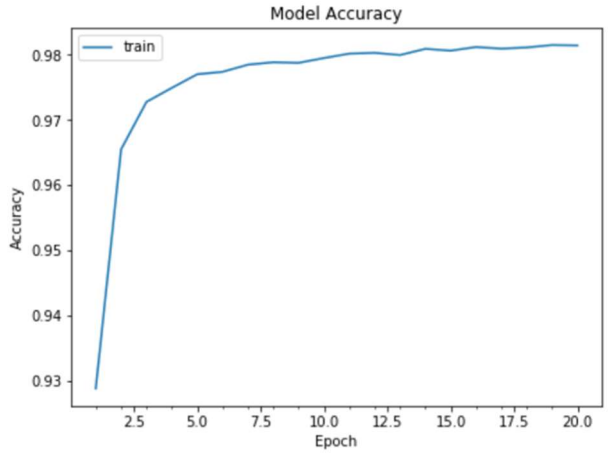


Figure 1. Keras model accuracy for Experiment 2

65. CONCLUSION & FUTURE WORK

The identification of money laundering cases is essential to prevent terrorist financing and drug and human trafficking. Many institutions have turned to machine learning as an approach to combat financial crime. This paper presented deep learning-based and classification-based machine learning methods to classify financial transactions. After applying different machine learning algorithms on both datasets, the deep learning Keras model and Random Forest Classifier model have shown a high accuracy number with promising results in reducing the number of false positives. Notably, the Random Forest Classifier gives high accuracy results on both datasets. While no best machine learning algorithm works for every problem, the deep learning Keras and Random Forest Classifier can be considered useful machine learning algorithms to apply to help improve AML solutions. Moreover, since I have not found any public research using the "feature_importances_" to determine what are the important features in detecting money laundering activities, I hope that the usage of this feature in this paper will contribute to the money laundering detection combat.

Future Work: With the lack of available datasets, one of the experiments in this paper used a synthetic financial dataset. There are some disadvantages when using synthetic data, such as the data might contain some biased information. If I have additional time and the condition allowed, I would like to tackle this money laundering problem with a real-world dataset. Also, if I have additional time, I will explore the graph convolutional neural network method for the Bitcoin graph dataset and some unsupervised machine learning methods on both datasets. Since with unsupervised machine learning methods, the system could identify the patterns that might be considered money laundering activities and suggest a new way to analyze the data through the interactions with uncategorized data. Therefore, I want to look at the effectiveness of detecting money laundering cases through unsupervised machine learning methods.

66. ACKNOWLEDGMENTS

I would like to express my gratitude to Dr. M Afzal Upal, my professor and academic advisor, for his guidance, support, and

valuable advice during the two years of my program, especially during the process of writing this paper. I also would like to send my appreciation to all professors at Mercyhurst University's Computing & Information Science Department for their informative lectures, which allowed me to gain the skills and expand my knowledge.

67. REFERENCES

[55] Lilley, P. 2000. Dirty Dealing: The Untold Truth about Global Money Laundering, International Crime and Terrorism. Kogan Page Publishers.

[56] Langdon, S. 2019. What is Money Laundering ? [How the Scheme Works & Examples]. Available at https://www.moneytaskforce.com/money/what-is-money-laundering/

[57] Genzman, L. 1997. Responding to Organized Crime: Laws and Law Enforcement. *In H.Abadinsky (Ed.) Organized Crime.* Belmont, CA, 342.

[58] McDowell, J. and Novis, G. 2001. The Consequences of Money Laundering and Financial Crime. *The Fight against Money Laundering: Economic Perspectives: An Electronic Journal of the U.S. Department of State.* Bureau of International Narcotics and Law Enforcement Affairs, U.S. Department of State. 6, 2, 6-8.

[59] PwC United States. Financial Crime Risk. Available at https://www.pwc.com/us/en/library/risk-regulatory/financial-crime-risk.html?

[60] United Nations Office on Drugs and Crime (UNODC). Illicit money: how much is out there? Available at http://www.unodc.org/unodc/en/frontpage/2011/October/illicit-money_-how-much-is-out-there.html

[61] Financial Action Task Force (FATF). Money Laundering: How much money is laundered per year? Available at https://www.fatf-gafi.org/faq/moneylaundering/

[62] Institute of International Finance. Machine Learning in Anti-Money Laundering – Summary Report. Available at https://www.iif.com/portals/0/Files/private/32370132_iif_machine_learning_in_aml_-_public_summary_report.pdf

[63] Global Financial Integrity. Money Laundering. Available at https://gfintegrity.org/issue/money-laundering/

[64] Canellis, D. 2018. Here's how criminals use Bitcoin to launder dirty money. *The Next Web*. Available at https://thenextweb.com/hardfork/2018/11/26/bitcoin-money-laundering-2/

[65] Weber, M., Domeniconi, G., Chen, J., Weidele, D. K. I., Robinson, T., and Leiserson, C. E. 2019. Anti-Money Laundering in Bitcoin: Experimenting with Graph Convolutional Networks for Financial Forensics. *KDD '19 Workshop on Anomaly Detection in Finance,* Anchorage, AK, USA.

[66] Canning, K. 2019. Anti-Money Laundering: Better with Data Science. *Business Data Partners.* Available at https://www.businessdatapartners.com/anti-money-laundering-data-science/

[67] Salehi, A., Ghazanfari, M., and Fathian, M. 2017. Data Mining Techniques for Anti Money Laundering. *International Journal of Applied Engineering Research.* 12, 20, 10084-10094.

[68] Senator, T. E., Goldberg, H. G., Wooton, J., Cottini, M. A., Khan, A. F. U., Klinger, C. D., Llamas, W. M., Marrone, M. P., and Wong, R. W. H. 1995. The FinCEN Artificial Intelligence System: Identifying Potential Money Laundering from Reports of Large Cash Transactions.

[69] Panigrahi, S., Sural, S., and Majumdar, A. K. 2009. Detection of Intrusive Activity in Databases by Combining Multiple Evidences and Belief Update. *IEEE Symposium on Computational Intelligence in Cyber Security.*

[70] Khan, N. S., Larik, A. S., Rajput, Q., and Haider, S. 2013. A Bayesian Approach for Suspicious Financial Activity Reporting. *International Journal of Computers and Applications.* 35, 4.

[71] Rajput, Q., Khan, N. S., Larik, A. S., and Haider, S. 2014. Ontology Based Expert- System for Suspicious Transactions Detection. *Computer and Information Science*. 7, 1.

[72] Wang, S. N. and Yang, J. G. 2007. A Money Laundering Risk Evaluation Method Based On Decision Tree. *The 6th International Conference on Machine Learning and Cybernetics,* 1, 283-286.

[73] Lopez-Rojas, E. A. and Axelsson, S. 2012. Money Laundering Detection using Synthetic Data. *The 27th annual workshop of the Swedish Artificial Intelligence Society (SAIS),* Örebro, Sweden. Published by Linköping University Electronic Press.

[74] Liu, R., Qian, X., Mao, S., and Zhu, S. 2011. Research On Anti-Money Laundering Based On Core Decision Tree Algorithm. *2011 Chinese Control and Decision Conference (CCDC)*, Mianyang, 4322-4325.

[75] Tang, J. and Yin, J. 2005. Developing an Intelligent Data Discriminating System of Anti-Money Laundering Based On SVM. *2005 International Conference on Machine Learning and Cybernetics,* Guangzhou, China, 6, 3453-3457.

[76] Keyan, L. and Tingting, Y. 2011. An Improved Support-Vector Network Model for Anti-Money Laundering. *2011 Fifth International Conference on Management of e-Commerce and e-Government*, Hubei, 193-196.

[77] Lv, L-T., Ji, N., and Zhang, J-L. 2008. A RBF Neural Network Model for Anti-Money Laundering. 2008 International Conference on Wavelet Analysis and Pattern Recognition, Hong Kong, 209-215.

[78] Zhu, T. An Outlier Detection Model Based on Cross Datasets 2006. Comparison for Financial Surveillance. *2006 IEEE Asia-Pacific Conference on Services Computing (APSCC'06)*, Guangzhou, Guangdong, 601-604.

[79] Cao, D.K. and Do, P. 2012. Applying Data Mining in Money Laundering Detection for the Vietnamese Banking Industry. In: Pan JS., Chen SM., Nguyen N.T. (eds) Intelligent Information and Database Systems. *ACIIDS 2012.* Lecture Notes in Computer Science, 7197, Springer, Berlin, Heidelberg.

[80] Synthetic Financial Datasets for Fraud Detection. Available at https://www.kaggle.com/ntnu-testimon/paysim1

[81] Elliptic Data Set. Elliptic, www.elliptic.co. Available at https://www.kaggle.com/ellipticco/elliptic-data-set

About the authors:

Huyen Vu is a graduate student at Mercyhurst University's Computing & Information Science Department.

How Does Mental Health Affects Unemployment? The Mediation Effect of Concentration Ability*

Chuhan Ouyang
Green Hope High School
Cary North Carolina United States
chuhan.ouyang@gmail.com

ABSTRACT

Compromised mental health severely torments residents across the US by creating concentration difficulties and reducing their competitiveness as employees. The purpose of this study was to evaluate the negative effects of mental health on employment and explore whether the effect is partially or fully mediated through the effect of concentration ability.

The data was obtained in a Behavioral Risk Factor Surveillance System survey, in which 178, 242 US adult residents reported their mental health condition, concentration ability, employment status, as well as other confounding variables such as race, age, and marital status. Logistic regression was employed to assess the association between mental health and employment status. Mediation analysis was used to test if the effect is partially or fully mediated through the effect of concentration problem.

Logistic regression analysis revealed that those with compromised mental health were 77% times as likely as those with good mental health to be employed. Mediation analysis showed that 34% of the effect of mental health on employment was mediated through concentration ability. Furthermore, the p-value for ACME in the mediation analysis was less than 0.001, indicating a statistically significant mediation.

Overall, there was a negative correlation between compromised mental health and employment status. A fairly large proportion of the effect could be explained by concentration problems. The findings validated the importance of future research and implementation of medical treatments for improvements in both health and employment status.

KEYWORDS

Mediation Analysis, Logistic Regression, Mental Health, Unemployment

1 Introduction

Our world has become so fast-paced, so intolerant of blunders, and so demanding that immense stress from school and work falls down upon our shoulders. Despite the ever-advancing medical technology, mental health has become a severe issue, tormenting daily lives of residents across the U.S. In fact, according to the Substance Abuse and Mental Health Services Administration, one in five US adults experience mental illness each year.

By definition, mental health is "a state of well-being in which every individual realizes his or her own potential, can cope with the normal stresses of life, can work productively and fruitfully, and is able to make a contribution to his or her community."[3]

Maintaining such a healthy state is vital because compromised mental health not only leads to physical illnesses such as cardiovascular disease and diabetes but also concentration difficulties marked by inattention and hyperactivity.[8]

The hallmark of concentration difficulty is attention-deficit hyperactivity disorder (ADHD), which severely impacts life. It creates troubles for patients to pay attention in work and complete assigned tasks, making them less competent as employees.[6]

Therefore, those who suffer from concentration difficulties are more likely to experience unemployment, which negatively affects the individual's life in various aspects. For instance, an unemployed person experiences low standard of living, insecurity of income, devalue self-esteem, and a shrinking social network.[7]

As seen above, mental health, concentration ability, and employment status are related. Such associations have been found by previous studies. For instance, a National Institute of Health study claims that mental illness impairs concentration ability.

In that study, students with mental illness lose the ability to learn effectively and fall further and further behind in school.[1] Another study, commissioned by *WISE Employment*, a non-profit organization that empowers job seekers to find meaningful work, reported that mental illness remains a serious obstacle to employment even in today's enlightening society.[4] As evident in those studies, there exists intricate links between these three variables: mental health, concentration ability, and employment.

• 1.1 Objective

In this study, I evaluated the effect of mental health on employment status using nationally representative data of adults in the United States. Meanwhile, I explored if the effect partially or fully mediated through the effect of concentration ability.

2 Study Methods

• 2.1 Data

Data from the Behavioral Risk Factor Surveillance System (BRFSS, website: https://www.cdc.gov/brfss/index.html) were used. 10BRFSS is a nation-wide telephone survey that collects data about U.S. residents regarding their health-related risk behaviors, chronic health conditions, and use of preventive services.

It now collects data in all 50 states as well as the District of Columbia and three U.S. territories, and completes more than 400,000 adult interviews each year. It is the largest continuously conducted health survey system in the world.

The study samples were limited to those aged 24-65 years old, and those who reported student or retired status were excluded.

• 2.2 Definition of Outcome

In BRFSS, participants were asked about their current employment status. Response options included the following:

1	Employed for wages
2	Self-employed
3	Out of work for more than 1 year
4	Out of work for less than 1 year
5	A homemaker
6	A student
7	Retired
8	Unable to work

In this study, those who chose 6 or 7 or 8 were excluded. Those choosing 1 or 2 or 5 were categorized as "employed" while those choosing 3 or 4 were grouped as "not employed."

Compromised mental health

In the survey, participants were asked "Now thinking about your mental health, which includes stress, depression, and problems with emotions, for how many days during the past 30 days was your mental health not good?" The variable is coded as 1 if they answered any value between 1 and 30, and coded as 0 if they answered none.

Concentration difficulty

Participants were asked "Because of a physical, mental, or emotional condition, do you have serious difficulty concentrating, remembering, or making decisions?" The variable is coded as 1 if they answered yes and 0 if they answered no.

Other variables

- Age group, as a continuous variable

2	25 <= AGE <= 29
3	30 <= AGE <= 34
4	35 <= AGE <= 39
5	40 <= AGE <= 44
6	45 <= AGE <= 49
7	50 <= AGE <= 54
8	55 <= AGE <= 59
9	60 <= AGE <= 64

- Gender
- Education:

1	Never attended school or only kindergarten
2	Grades 1 through 8 (Elementary)
3	Grades 9 through 11 (Some high school)
4	Grade 12 or GED (High school graduate)
5	College 1 year to 3 years (Some college or technical school)
6	College 4 years or more (College graduate)

- Income
- Marital status: a binary variable "married/partnered" was created with yes/no values
- Race and ethnicity

1	White only, non-Hispanic
2	Black only, non-Hispanic
3	American Indian or Alaskan Native only
4	Asian only, non-Hispanic
5	Native Hawaiian or other Pacific Islander only, Non-Hispanic
6	Other race only, non-Hispanic
7	Multiracial, non-Hispanic
8	Hispanic

- Physical health problem: a binary variable "have physical health problem" was created with yes/no values
- Activity limitation : a binary variable "have activity limitation" was created with yes/no values

• 2.3 Analysis

I hypothesized that mental health affects concentration ability, which is associated with employment status.

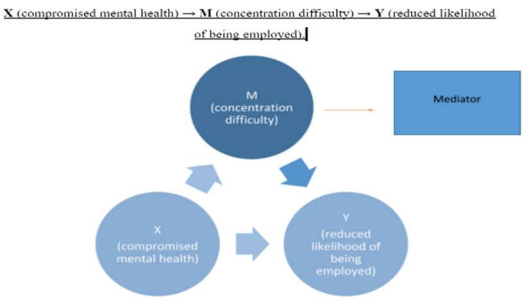


Figure 1: Independent, dependent variables and the mediator

In other words, the concentration problem is a mediator that (partly) explains the underlying mechanism of the relationship between mental health and employment status.

In general, mediation analysis involves four steps which is shown in the flowchart below [1]:

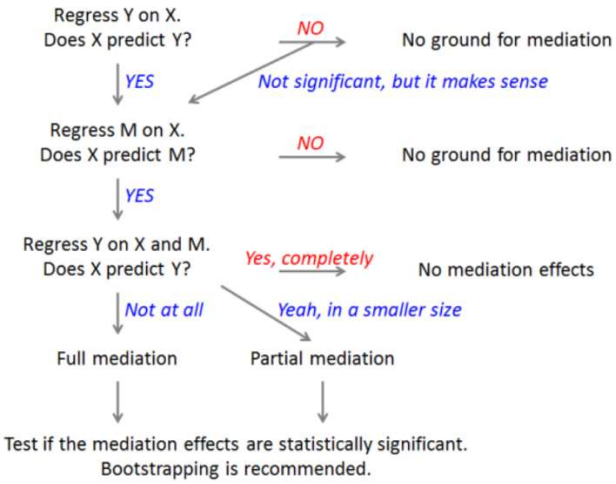


Figure 2: Steps of mediation analysis, adapted from [2]

A detailed explanation of each step is as below:

Step 1. To test the relationship between X and Y

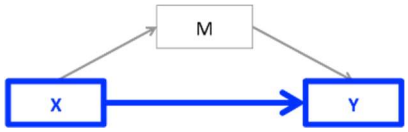


Figure 3: Regression model using X to predict Y

The model is:

$$Y = b0 + b1X + b3 * any\ potential\ confounders$$

A statistically significant relationship between X and Y is expected, meaning that a b1 with a P-value <0.05 is expected.

Step 2. To test the relationship between X and M

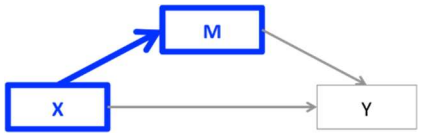


Figure 4: Regression model using X to predict M

The model is:

$$M = b0 + b2X$$

A mediation makes sense only if X affects M. Therefore, I expect to see a statistically significant relationship between X and M, meaning that a b2 with a P-value <0.05 is expected.

Step 3. To test the effect of X on Y after including M

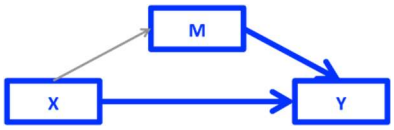


Figure 5: Regression model using X and M to predict Y

The model is;

$$Y = b0 + b3X + b4M$$

Or another way to understand it:

$Y = b0 + b3X$ (X can impact Y through all possible pathways, including M and other mediators)

$Y = b0 + b3X + b4M$ (the effect of X here is the remaining effect of X after taking into account its effect through M)

If a mediation effect exists, when M is added in the regression, the effect of X on Y will disappear or weaken. In other words, I expect the following:

1. M to affect Y
2. X to no longer affect Y, or X to still affect Y but in a smaller magnitude.

Therefore, I would expect a significant b_4, and b_3 to be smaller than b1.

Step 4. To test if the mediation effect is statistically significant

If a mediation effect is seen from the previous steps, I further test if this mediation effect is statistically significant, using a bootstrapping approach.[2]

Steps 1 to 3 were achieved through Logistic regression analysis. It is a type of generalized linear regression for analyzing the relationship between a set of explanatory variables and a binary outcome variable. For example, in step 1, the outcome is if a person is employed.

The general formula of logistic regression is:

$$\ln(odds\ of\ an\ event\ occuring) = \ln\left(\frac{p}{1-p}\right) = B + B1 * X1 + B2 * X2 + \cdots + Bn * Xn$$

P is the probability of an event, which is convertible with odds.

Xn is a predictor variable, and n is a regression coefficient. The relationship between the odds ratio and the coefficients are:

$OR = e^B$

If the coefficient β of a variable *Xn* is larger than 0, *Xn* is related to a higher odds/probability of the event. The odds ratio related to *Xn* is above 1 in this case.

- If the coefficient of a variable *Xn* is equal to 0, Xn is not related to the event. The odds ratio related to Xn is equal to 1 in this case.

- If the coefficient of a variable Xn is smaller than 0, *Xn* is related to a lower odds/probability of the event. The odds ratio related to Xn is below 1 in this case.

All analysis was performed in R software 3.6.1.

3 Results

• 3.1 Profile of Participants

The final study sample includes 178,242 participants.

Of all participants, 47% were males, and 53% were females. Frequency of age group:

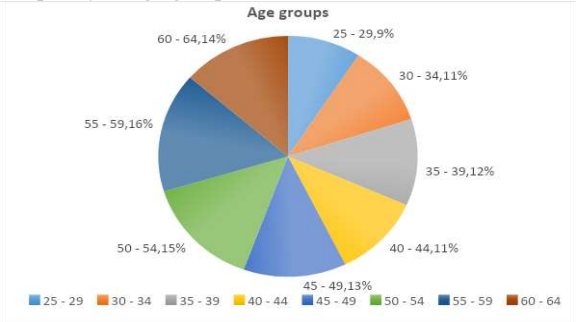


Figure 6: Pie chart of age groups of participants

Frequency of races:

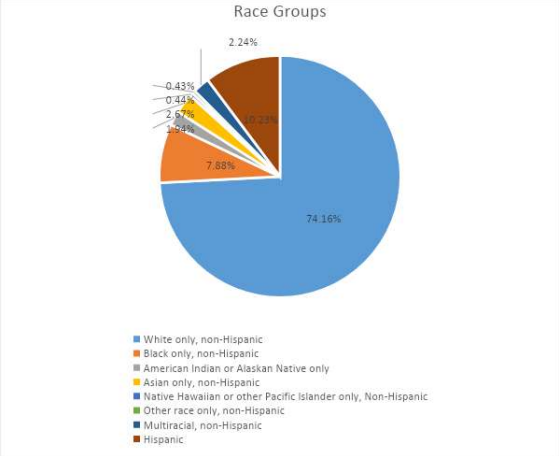


Figure 7: Pie chart of race of participants

93.7% were employed and 6.3% were unemployed. 34.6% reported having poor mental health on one or more days in the past 30 days.

• 3.2 Bivariate Analysis Results

Firstly, when looking at bivariate analysis across mental health status, employment status, and concentration difficulty, I found that they are related with each other. Those with poorer mental health were more likely to have concentration difficulties and less likely to be employed. Those with concentration difficulties were less likely to be employed.

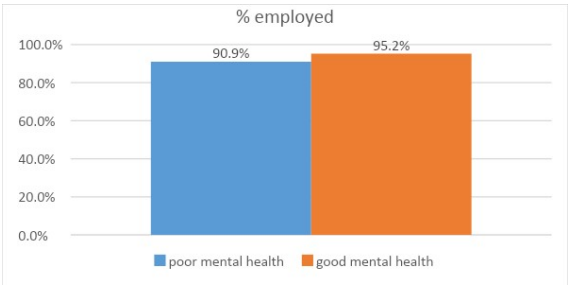


Figure 8: Bivariate Analysis of mental health and employment status

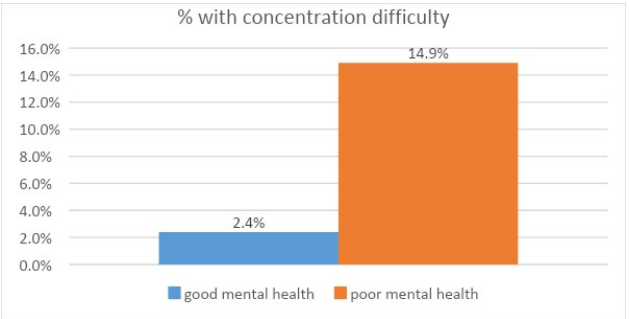


Figure 9: Bivariate Analysis of mental health and concentration difficulty

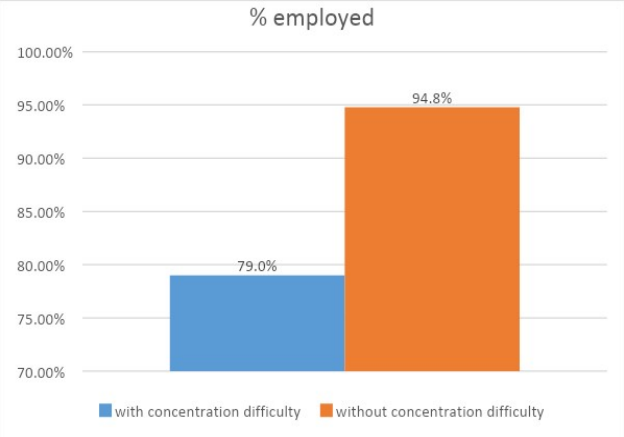


Figure 10: Bivariate Analysis of employment and concentration difficulty

• 3.3 Mediation Analysis

Step 1. To test the relationship between X and Y

Results from multivariate Logistic regression are as below:

Logistic regression predicting employment status, before including concentration difficulty in the model								
	Estimate	Std. Error	z value	Pr(>|z|)		Odds Ratio	lower CI	Upper CI
(Intercept)	0.9816	0.05151	19.058	<0.0001	***	2.6689	2.4129	2.952
poor mental health	-0.2583	0.02353	-10.98	<0.0001	***	0.7723	0.7375	0.808
physical problem	-0.1416	0.02501	-5.662	<0.0001	***	0.8679	0.8264	0.911
limited_activities	-0.6622	0.02733	-24.223	<0.0001	***	0.5156	0.4887	0.544
age group	-0.0702	0.00469	-14.963	<0.0001	***	0.9321	0.9236	0.94
male	-0.2486	0.02138	-11.629	<0.0001	***	0.7798	0.7478	0.813
education	0.0001	0.01185	0.015	0.98834		1.0001	0.9771	1.023
income	0.7339	0.00844	86.861	<0.0001	***	2.0832	2.0491	2.118
married/partnered yes vs no	0.1123	0.02285	4.914	<0.0001	***	1.1188	1.0698	1.17
Race: Race1 as reference								
RACE2	-0.3095	0.03253	-9.514	<0.0001	***	0.7337	0.6886	0.782
RACE3	-0.3536	0.05474	-6.46	<0.0001	***	0.7021	0.6311	0.782
RACE4	-0.0872	0.07079	-1.233	0.21776		0.9164	0.7993	1.055
RACE5	-0.3202	0.122	-2.625	0.00867	**	0.7259	0.5744	0.927
RACE6	-0.163	0.1404	-1.161	0.24565		0.8495	0.6497	1.127
RACE7	-0.1746	0.06321	-2.762	0.00574	**	0.8397	0.7429	0.951
RACE8	0.2991	0.03355	8.913	<0.0001	***	1.3486	1.2631	1.44

Figure 11: Logistic Regression Results, using X to predict Y

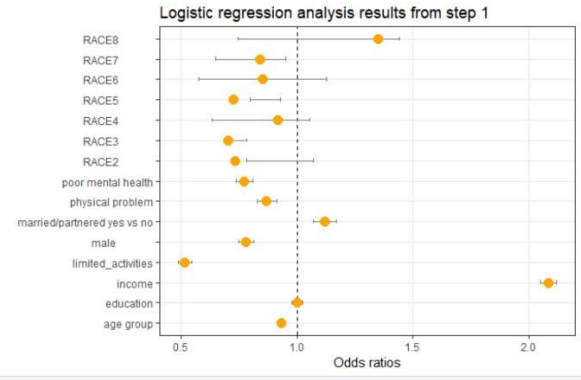


Figure 12: Odds Ratio of Independent Variables from step 1

Results showed that the Odds Ratio for poor mental health is below 1, meaning that poor mental health is related with less likelihood of being employed.

The p-value for this regression coefficient is less than 0.001, indicating a statistically significant relationship.

Step 2. To test the relationship between X and M

Logistic regression predicting concentration difficulty								
	Estimate	Std. Error	z value	Pr(>|z|)		Odds Ratio	lower CI	Upper CI
(Intercept)	-1.473	0.052	-28.094	<0.0001	***	0.229	0.206	0.253
poor mental health	1.437	0.024	59.346	<0.0001	***	4.21	4.016	4.416
physical problem	0.367	0.023	15.697	<0.0001	***	1.444	1.379	1.512
limited_activities	1.063	0.023	44.59	<0.0001	***	2.896	2.764	3.035
age group	-0.025	0.004	-5.538	<0.0001	***	0.974	0.966	0.983
male	0.008	0.021	0.415	0.67821		1.008	0.967	1.051
education	-0.307	0.011	-26.06	<0.0001	***	0.735	0.718	0.752
income	-0.348	0.008	-42.624	<0.0001	***	0.705	0.694	0.717
married/partnered yes vs no	-0.205	0.022	-9.158	<0.0001	***	0.814	0.779	0.851
Race: Race1 as reference								
RACE2	-0.203	0.037	-5.378	<0.0001	***	0.815	0.756	0.878
RACE3	-0.038	0.063	-0.609	0.54239		0.962	0.849	1.087
RACE4	-0.343	0.084	-4.05	<0.0001	***	0.709	0.598	0.834
RACE5	-0.186	0.146	-1.277	0.20161		0.829	0.617	1.096
RACE6	0.43	0.132	3.245	0.00117	**	1.537	1.178	1.983
RACE7	0.325	0.056	5.717	<0.0001	***	1.384	1.236	1.545
RACE8	-0.13	0.033	-3.932	<0.0001	***	0.877	0.822	0.936

Figure 13: Logistic Regression Results, using X to predict M

Results showed that the Odds Ratio for poor mental health is above 1, meaning that poor mental health is related with higher likelihood of concentration difficulty.

The p-value for this regression coefficient is less than 0.001, indicating that a statistically significant relationship.

Step 3. To test the effect of X on Y after including M

Logistic regression predicting employment status, after including concentration difficulty in the model								
	Estimate	Std. Error	z value	Pr(>|z|)		Odds Ratio	lower CI	Upper CI
(Intercept)	1.116	0.052	21.354	<0.0001	***	3.055	2.757	3.385
poor mental health	-0.174	0.024	-7.195	<0.0001	***	0.839	0.8	0.88
concentration difficulty	-0.567	0.03	-18.916	<0.0001	***	0.566	0.534	0.601
physical problem	-0.128	0.025	-5.098	<0.0001	***	0.879	0.837	0.924
limited_activities	-0.578	0.027	-20.694	<0.0001	***	0.56	0.53	0.592
age group	-0.071	0.004	-15.265	<0.0001	***	0.93	0.921	0.939
male	-0.25	0.021	-11.689	<0.0001	***	0.778	0.746	0.811
education	-0.015	0.011	-1.303	0.1925		0.984	0.961	1.007
income	0.715	0.008	83.852	<0.0001	***	2.044	2.01	2.078
married/partnered yes vs no	0.105	0.022	4.59	<0.0001	***	1.111	1.062	1.162
Race: Race1 as reference								
RACE2	-0.33	0.032	-10.121	<0.0001	***	0.718	0.674	0.766
RACE3	-0.369	0.054	-6.723	<0.0001	***	0.691	0.62	0.77
RACE4	-0.1	0.07	-1.422	0.15489		0.904	0.788	1.041
RACE5	-0.349	0.122	-2.859	0.00425	**	0.705	0.557	0.9
RACE6	-0.136	0.141	-0.963	0.33549		0.872	0.666	1.16
RACE7	-0.157	0.063	-2.468	0.01359	*	0.854	0.755	0.969
RACE8	0.279	0.033	8.295	<0.0001	***	1.321	1.237	1.412

Figure 14: Logistic Regression Results, using X and M to predict Y

When including both poor mental health and concentration problems in the model, the coefficient for X is -0.174 , which is smaller in magnitude than in step 1. This is a good sign showing that adding M reduces the effect of X.

Step 4. To test if the mediation effect is statistically significant

	Estimate	95% CI Lower	95% CI Upper	p-value
ACME (control)	-0.00446	-0.00484	0.00	<2e-16
ACME (treated)	-0.00489	-0.00526	0.00	<2e-16
ADE (control)	-0.00879	-0.01056	-0.01	<2e-16
ADE (treated)	-0.00921	-0.01105	-0.01	<2e-16
Total Effect	-0.01368	-0.01544	-0.01	<2e-16
Prop. Mediated (control)	0.32631	0.26170	0.40	<2e-16
Prop. Mediated (treated)	0.35752	0.29215	0.43	<2e-16
ACME (average)	-0.00468	-0.00504	0.00	<2e-16
ADE (average)	-0.00900	-0.01080	-0.01	<2e-16
Prop. Mediated (average)	0.34191	0.27693	0.42	<2e-16

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Significance codes: ***<0.001 **< 0.01 *<0.05

Figure 15: Bootstrapping approach results of mediation

In the output, ACME (Average Causal Mediation Effects) and "Prop. Mediated (average)" are the two major metrics to look at. ACME has a p-value less than 0.001, meaning that it is different from zero which further means that the mediation is statistically significant.

"Prop. Mediated (average)" is the proportion of the effect of X on Y that is explained by M. It can be seen that 34% of the effect of mental health on employment is explained by concentration ability.

4 Discussion

4.1 Discussion of Study Methods

In this study, I used the method of mediation analysis with confounding effects, which is commonly used in psychological research. In fact, the mediation analysis guided by Baron & Kenny is one of the most frequently cited papers in psychological literature.[5]

Since the aim of this study is to gain a fundamental understanding of the underlying mechanism between unemployment and mental health, mediation analysis is the best approach. I also considered confounding effects of variables such as age, gender, and race because of the complicity of the situation.

Mediation is oftentimes confused with confounding, since both measure how the relationship between an independent and dependent variable varies once a third variable is added. However, the two effects refer to very different concepts.

For mediation, researchers explore whether a third variable (partially) explains the underlying mechanism of the relationship between the independent variable and dependent variable. As seen through the figure below, there are two paths in a mediation model. The independent variable can directly affect the dependent variable. Or, the independent variable can indirectly affect the dependent variable through a mediator.

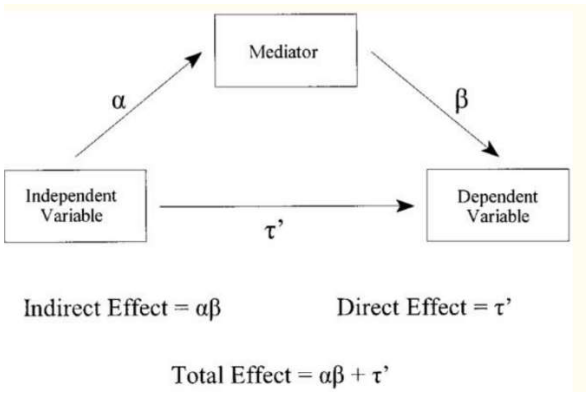


Figure 16: Illustration of the mediation analysis model

On the other hand, a confounder is a third variable that is related to both the independent and dependent variable and obscures such a relationship. If compared with the figure above, a confounding model actually reverses the direction of the effect between the independent variable and the third variable.

Overall, mediation provides useful insights about the relationship of the two factors of interests while confounders produce spurious effects. In this study, concentration ability is a mediator, not a confounder, because it explains the mechanism through which mental health affects unemployment. This information is useful and significant.

To analyze the mediation effect and its statistical significance, I chose the bootstrapping method because its nonparametric approach of hypothesis testing does not assume the shape of distribution of variables, unlike the Sobel test which assumes normal distribution.

Introducing asymmetry and non-normality, the bootstrapping approach circumvents statistical problems. However, it is worth noting that this quantitative analysis only lends support to the hypothesis that mental health negatively impacts unemployment with partial mediation on concentration ability. It is not probable to assume that mental health cause unemployment.[6]

4.2 Discussion on Findings

Using the method mentioned above, my finding is that poor mental health is negatively related to employment status. The effect partly goes through concentration problems. The proportion of mediated effect is not very high but is statistically significant.

Other researchers have made similar conclusions. For instance, the Institute for Work and Health reveals that the cause-and-effect relationship can work in both directions: not only will mental health problems make it more difficult for a person to obtain and/or hold a job, but unemployment also may worsen mental health.

In other words, mental illness and unemployment forms a negative cycle, and those who are trapped in it suffer immensely.

Therefore, in order to stop the negative cycle, it is paramount to improve mental health for the susceptible population and provide career assistants in their job searching.

To enhance mental health conditions, professional medical assistance is the optimal way. In a National Alliance on Mental Illness study, it is found that 90% of patients with mental illness receive considerable reductions in their symptoms after seeking professional help.[4]

Making such helpful treatments widespread and affordable should therefore be one of the focus of national funding. Meanwhile, all working units should have at least one trained therapist to ensure healthy mental conditions of all employees.

Since concentration ability is proven to be a statistically significant mediator in the relationship, psychologists workers who are helping the mentally compromised population to find jobs can focus on improving the patients' concentration ability. For example, they can prescribe medications and use brain exercises to improve concentration ability of the patients during treatment.

With professional assistance, patients will be more likely to get employed and boost their self-confidence, ending the cycle of unemployment and mental illness.

Besides improving mental health and concentration ability, another crucial step to help patients is to remove their stigma, which significantly impedes employment. In fact, according to an Australian study commissioned as part of Mental Health Week, "the stigma of mental illness often has a greater impact on people's employment prospects than physical disability or illness."[4]Negative descriptions of mental illness are present everywhere: social media, TV shows, dramas.

Patients are being characterized as insane, malevolent, and dangerous. Contrary to common belief, these absurd prejudices are mostly erroneous. A WISE Employment research showed that 72% of small and medium enterprises with a mentally-ill employee had a positive experience.[4] To combat these wrong stereotypes, advocacy groups can actively protest. Harnessing the power of social media, such groups can spread stories of those suffering from mental illness "holding jobs, providing for themselves, and living as good neighbors in a community".[1]

In the future, researchers can further focus on techniques of improving mental health, concentration abilities, and employment status. Quantitative analysis models can be used in investigations. For instance, future researchers can use logistic regression models to test to what extent would introducing therapists to company or demystifying and normalizing mental illness on social media improves the life qualities of the susceptible population.

In addition, since concentration difficulties explain only 34% of the effect of mental health on unemployment, future researchers can investigate other potential mediators through mediation analysis. For example, stigma, as mentioned above, is shown to have a great impact on patients' employment status. Exploring other mediators allows researchers to further understand the underlying mechanism of the relationship between mental health and unemployment.

4.3 Limitations and Strengths

There are limitations of the study based on the way the survey was conducted. The variables are self-reported. So, there may be response bias, which means participants are not answering questions untruthfully. They may feel pressure to give answers that are socially acceptable.

For example, someone suffering from mental illness may hide their true conditions. In addition, there is also non-response bias. The participants selected to complete the survey may fail to answer the telephone and complete all survey questions.

To mitigate non-response bias, the BRFSS employs a statistical method called post stratification to weight the survey data and adjust each respondent data to known proportions of gender, region, race, and other characteristics.14The weighting makes the sample more representative of the national population and adjusts for the bias.

Moreover, the large sample size of the study ensures the accuracy of the conclusion. There are more than 400,000 participants across all 50 states in the US. The large and nationally representative sample reduces the margin of error. In addition, the study uses the stringent statistical test of mediation to examine the relationships between variables. Inclusion of multiple potential confounders in the logistic regression model controls for spurious effects.

5 Conclusion

Overall, logistic regression models showed that compromised mental health is negatively related to employment status. Specifically, those with mental health problems are 23% less likely to be employed than healthy individuals. A fairly large proportion of the effect could be explained by concentration problems.

The findings validate the importance of improvements in both health and employment status. In this fast-paced, intolerant, and demanding world, future research and prompt implementations of medical treatments are the best ways to help the susceptible population escape from the negative cycle of unemployment and self-devaluation.

REFERENCES

[1] Bethesda. Information about Mental Illness and the Brain. *National Institutes of Health* (2007). Available at: https://www.ncbi.nlm.nih.gov/books/NBK20369/.

[2] Introduction to Mediation Analysis. *University of Virginia Library.* Available at: https://data.library.virginia.edu/introduction-to-mediation-analysis/.

[3] Mental Health: A State of Well-Being. *World Health Organization* (2014). Available at: https://www.who.int/features/factfiles/mental_health/en/.

[4] Nordqvist, C. Mental Illness Affects Job Prospects More Than Physical Disability. *Medicalnewstoday* (2012). Available at: https://www.medicalnewstoday.com/articles/251237.php.
[5] Preacher, Kristopher J., and A. F. H. SPSS and SAS Procedures for Estimating Indirect Effects in Simple Mediation Models. *Behav. Res. Methods, Instruments, Comput.* 36, 717–731 (2004).
[6] Stöppler, Melissa Conrad. Concentration Problems: Check Your Symptoms and Signs. *MedicineNet* (2019). Available at: https://www.medicinenet.com/difficulty_concentrating/symptoms.htm.
[7] Unemployment and mental health. *IWH* (2009). Available at: https://www.iwh.on.ca/summaries/issue-briefing/unemployment-and-mental-health.
[8] 10 Facts on Mental Health. *World Health Organization* (2014) Available at: https://www.who.int/features/factfiles/mental_health/mental_health_facts/en/index4.html.
[9] Key Substance Use and Mental Health Indicators in the United States: Results from the 2018 National Survey on Drug Use and Health. *Substance Abuse and Mental Health Services Administration* (2018). Available at: Key Substance Use and Mental Health Indicators in the United States: Results from the 2018 National Survey on Drug Use and Healt.
[10] 2017 Survey Data Information. *Behavioral Risk Factor Surveillance System* (2017). Available at: https://www.cdc.gov/brfss/annual_data/annual_2017.html
[11] Szumilas, Magdalena. Explaining Odds Ratios. *National Institute of Health* (2010). Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2938757/
[12] MacKinnon, David P. et al. Equivalence of the Mediation, Confounding and Suppression Effect. *National Institute of Health* (2010). Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2819361/.
[13] Nichols, Hannah. Five ways to boost concentration. *Medical News Today* (2017). Available at: https://www.medicalnewstoday.com/articles/320165

Seeking Mental Health Treatment Using Machine Learning

Tiffany Gonzalez
Department of Computing & Information Science
Mercyhurst University
501 East 38th Street
Erie PA 16546
tgonza75@lakers.mercyhurst.edu

ABSTRACT

Mental health has been a popular topic in society and mental health research has expanded more rapidly in the last decade than ever before. Machine learning has increasingly been changing mental health research. For instance, machine learning is a valuable technique to help predict which individuals may be facing a mental health crisis. In this study, logistic regression, decision tree, and random forest methods will be used to compare which models performed the most accurately when determining if a person has sought mental health treatment.

Keywords

Mental Health, Employee Mental Health, Tech Industry, Logistic Regression, Decision Tree, Random Forest

1. INTRODUCTION

In the past few years, the phrases "mental health" and "mental illness" have been thrown around in the media more than ever. Sometimes, the two can be misinterpreted and used interchangeably. However, they are two different types of conditions. Mental illness is identified as a significant impairment in an individual's abilities. Abilities can be cognitive, affective, or relational [1]. Each year in the United States, one in five adults encounter mental illness, according to the Society of Human Resources. Mental illness is a stretch from mental health. If an individual does not take care of their mental health, a mental illness can prevail [2]. Severe mental illness disorders include bipolar disorder, severe depression, and schizophrenia. Mental illness disorders are diagnosed by certified physicians. On the other hand, mental health is the psychological state of an individual's well-being. Out of all Americans who are employed 18% reported having suffered symptoms of a mental health disorder in the month prior [3]. Knowing this suggests a coworker of yours could potentially be or are already negatively affected by their mental health. This is why everyone should be checking in on themselves and with a doctor to make sure they are mentally healthy.

Many people carry subconscious biases towards poor mental health. Biases toward those with mental health issues can lead to serious workplace conflicts. A workplace conflict for example could be discrimination towards those with a mental health condition. Additionally, those being discriminated may not open up about personal challenges at work because of fear that others will label the individual weak or incompetent. Incompetent in the sense that the individual may not be able to perform a task to the best of their ability due to their condition. Fear of being labeled weak or incompetent can drive the individual to perform their work duties poorly. Furthermore, fear of being discriminated can prevent the individual from seeking mental help. For physical problems or even physical check ins, an annual check-up is fairly common in all individuals. Blood pressure, temperature, weight and height are all things that are checked yearly to make sure a human is in a healthy condition, physically. If there are issues with their blood pressure, then the individual has blood drawn to check for thyroid issues, iron levels, or cholesterol. This type of appointment is commonly accepted in the workplace. Furthermore, it is accepted that when workers need to take half days to go to an annual physical appointment. In contrast, it is not common for every individual to go to a mental health professional for an annual mental health appointment. It is a common fact that if the individual were to schedule a mental health appointment, they may not tell their boss why they need the day off as they would for a physical health appointment. The worker may not even tell their co-workers why they will not be at work. Is mental health, something that each of us must take care of, unacceptable to talk about at work? How many people are comfortable talking about their mental health to their supervisors and co-workers? What is the difference compared to those with physical health? If mental health isn't as accepted, is there a way to make this more acceptable in the workplace?

Safety & Health reported in a poll about workplaces including mental health in the safety guidelines. The responses were closely divided in half with 51 percent responding "No" and 49 percent answering "Yes" [4]. Mental health is so prevalent in every individual's lifestyle. Every person has thoughts and each person has different ways of dealing with their thoughts during the day. Some things affect one individual more than the next individual. We all need some sort of coping mechanism to move through each of our own unique lives. Chosewood, director of the Office for Total Worker Health at the National Institute for Occupational Safety and Health said, "Depression and anxiety cross every industry and occupation, every socioeconomic status, every race and ethnicity" [4]. Depression and anxiety aren't just for blue collar workers, they can be found in CEO's as well. For example, Kate Spade, a multimillionaire fashion designer committed suicide in 2018 due to depression and anxiety [5]. Chosewood continued saying "Unlike other chronic conditions that usually don't start in workers until their 40s, 50s or even 60s, mental health concerns typically present in a worker's 20s or 30s and can

last throughout almost the entire working career" [4]. The mindset of the majority who responded no to including mental health in professional responsibility is what needs to be changed. Something so important, such as psychological well-being, should be looked at just as closely as a physical health condition.

Research on costly health conditions was reported. Depression ranked number one for most costly, and anxiety ranked fifth — with obesity, arthritis, and back and neck pain in between. The #1 and #5 most expensive health cost is a mental health condition [3]. Would knowing this make employers more conscious about what they believe are important and worth talking to their employees about? Would knowing this also make employers offer mental health benefits for their employees? Research conducted by the World Health Organization (WHO) found that workers with depression reported the equivalent of 27 lost workdays per year — nine of them because of sick days or other time taken out of work, and another 18 reflecting lost productivity [6]. The toll of the productivity loss significantly hurt the companies involved. Thus, it is hoped that an employer would see these numbers and take the necessary steps to help those with mental health conditions, such as offering benefit plans. By receiving mental health benefits, the comfort level and acceptance that employees would feel coming to work every day would increase productivity.

Adults spend roughly one-third of their lives at work [7]. Employees should be able to communicate to their boss that they don't mentally feel okay and the boss should be understanding in return. Today, when someone expresses that they've been diagnosed with a physical illness such as arthritis, no one says "You should just think positive if you want your muscles to work better". Or if they may express how it hurts, no one says "Quit exaggerating. We all have problems in life." So why are similar statements said to those individuals with poor mental health hear. None of us have exceptional days every day at work. Some individuals have the non-exceptional feeling worse than others.

Other studies [8,9,10] have found that policies implemented in the workplace and similar helpful and positive approaches are needed to support better mental health support in the workplace. This includes employers giving set guidelines on their responsibilities towards their employees, offer preparation classes to help coworkers identify and address mental health issues in other individuals, as well as financial incentives. Financial incentives for example can be employers integrating mental health benefits for their employees. Most people aren't ashamed to see a doctor to help them take care of their bodies, why should people feel ashamed to see a therapist to take care of their minds? Are people ashamed to take care of their mental health because benefits aren't always available at work, so they are not sure what their employer supports? For this questionable discrepancy, the following is the research question being tested in this paper: Can we predict if a person should be treated for his/her mental health condition or not according to the values obtained in the dataset? Knowing whether or not physical health is more accepted than mental health in the workplace will help us determine the solution to an individual needing care or not. If so, what can be done in the workplace to help the individual that needs treated for their mental health or mental illness? Thus, depending on the results, we can perhaps create a more open and acceptable work environment for those dealing with mental health related issues and get them the treatment they need.

2. RELEVANT WORK

Mental health may be just as widely accepted as physical health, and the stigma surrounding it may be wrong. There haven't been any studies comparing people's views on mental health in the workplace to physical health in the workplace. Comparison factors of the following relevant works to this paper include: the comfort level of talking to supervisor or coworkers, feeling supported or unsupported if one were to have a mental health condition, or not receiving benefits for mental health care like physical health care. In this section, we will look at relevant works that have addressed the question of how mental health is looked at in the workplace. None of the relevant studies reported in this section compare mental to physical health, which is what makes this paper so unique. In this paper, we will reach a conclusion to determine how people feel about mental health in their current work environment as well as determining whether a person should be treated for their mental health condition.

In one study [8], the objective was to look at mental health being a neglected aspect at work and because of this, how stress became a factor. Workplace stress can be cause by poor work organization and lack of support from coworkers and supervisors. Although this article doesn't compare mental health to physical health, it is a relevant work because questions asked in the dataset include topics on supervisor and employee relations. Unsupportive coworkers or supervisors can lead to stress in the workplace and thus can lead to someone needing treated for their mental health condition [8].

The study used data from 35 countries and showed if an individual faced discrimination for having a mental health issue. A cross-sectional study was conducted with analysis of variance and generalized linear mixed models were used to analyze the data. The results revealed 62.5% of people experienced discrimination for their mental health in the workplace. The study found that about two-thirds of employees that experienced depression faced discrimination at their current workplace or while applying for a new job [8]. Discrimination was shown to be prevalent in higher income countries compared to lower income countries. Women showed to have a higher risk of discrimination as well.

If organizations are made aware of this, they can encourage staff to seek appropriate mental health care as per need. This would not only lead to improved care for persons with mental health conditions, but it would also lead to a situation where employees are comfortable going to work regardless of their mental health condition. In addition, taking actions early on would prevent a mental health issue from turning into a mental illness.

Another relevant work [9] researched the number of employees who left work due to stress around their mental health condition in the workplace. Data was taken from different countries around the world to indicate the number of employees who did in fact

leave work. Work-related stress is a major cause of occupational ill health, poor productivity and human error [9]. This means due to sickness, employee turnover, and poor performance in duties at work have increased the absence at work ratio. Mental health issues have an impact on employers and businesses directly through those factors mentioned previously. In addition, they impact an employee's morale.

The result of this study found that out of all work-related disabilities in the Netherlands, 58% of them were associated to a mental health issue. In the UK, it is estimated that around 30-40% of employees absences due to sickness were connected to some form of mental health issue [9]. This shows that having a mental health condition or issue that is not supported by employers could have negative effects on the individual as well as the company. This analysis is relevant to this paper because if there were more health care options in place at work, then those suffering from a mental health condition could receive treatment.

The last relevant work [10] researched how to improve mental health in the workplace. The study addressed support for individuals with a mental health condition during their work-related stress. As mentioned previously, two mental health conditions, depression and anxiety, top the list of the most burdensome and costly illnesses in the United States. Over $200 billion a year is estimated to be used towards the two conditions. Furthermore, approximately one-third of the mental health cost burden is related to productivity losses at workplaces. This includes unemployment, disability, and the outcomes due to lower work performance [10].

In this study [10], employees with depression showed to be significantly less productive during their work week than those with mild or no depression episodes. Similar to the last relevant work, this analysis is relevant because the dataset involved includes questions about individual's productivity at work and whether or not they were being treated for their condition. The results showed a substantial amount of those with moderate (57%) or severe (40%) depression did not seek any treatment. One of the authors in this relevant work, Crighton, said "Supportive bosses are key to achieving health and wellness" [10].

Individuals that answer having supportive supervisors makes the stigma that mental health is not as accepted as physical health incorrect. If that stigma is incorrect within the tech industry, this may mean more people that need treated for a mental health condition will be treated due to benefits being offered. Nonetheless, this support for mental health is the hope for the future. Crighton continued saying "Toxic bosses lessen engagement, increase disability and workers' compensation claims, and negatively impact productivity" [10]. The mindset and stigma from toxic bosses in addition to coworkers is hoped to be reversed as mental health becomes more popular today than ever before.

3. PROPOSED SOLUTION

Some occupations are at a higher risk of mental health problems than others. In this paper, we look at employees in the tech industry, which is believed to have high stress because of daily changes and innovations. As stated previously, there have been no machine learning studies that compare physical health to mental health in the workplace. Furthermore, there have not been any studies of how that affects an individual's treatment while working. In order to conclude whether physical health is more accepted than mental health in the workplace, I need to use a dataset that encompasses answers to relevant questions from individuals in the workplace. Open Source Mental Illness has been a popular dataset that has previously been used in studies. The dataset from OSMI is very useful for the research question this paper proposes. As previously discussed, the results will include employees or employers' responses to questions concerning seriousness between mental and physical health in the workplace. Individuals will be separated by country, family history, age groups, and gender. The machine learning question to be answered is whether an individual sought treatment for his/her mental illness according to the values obtained on mental and physical health in the workplace. Additionally, the demographic factors I previously listed will provide more context for the results.

4. EVALUATION

4.1 Data Collection and Cleaning

The first step in the evaluation process is data collection. For this research paper, questions and answers relating to mental and physical health in the workplace were collected from the Open Sourcing Mental Illness (OSMI) dataset. OSMI collected data on this topic from 2014 to 2019. The first challenge was how to combine the OSMI datasets from 2014 to 2019. I decided that features in the datasets from 2014 and 2016 served a better purpose than 2017 to 2019. The reason being is there were not enough features and the way the questions were asked in comparison to all the years being combined. The next step in the evaluation process was to combine the 2014 dataset with the 2016 dataset. The 2016 dataset had 63 features while the 2014 dataset had 24 features. The 24 features were more than enough to complete the analysis and machine learning question I set out for in this paper. Thus, when I combined the 2014 and 2016 dataset, I ended with 23 similar features. I cleaned the data of 2014 separately than the data for 2016 due to the two having the same features but different wordings of the question. I changed all features to the same feature name. For example, in 2014 the feature name was "treatment" and in 2016 the feature name was "Have you ever sought treatment for a mental health issue from a mental health professional?". When combined in the 2014_2016 dataset, the feature name is "sought_treat". Next, I dropped all features that were not needed for the purpose of this research question and ended with 23 features total. Once both datasets were cleaned up, I exported them as csv files and combined them manually. I then read in the new combined dataset and began to chart to see the data and its relationships before I began to use machine learning with the data. I then scaled and fitted the age feature appropriately. Following, I split the data into train (70%) and test (30%). I created a classification model that evaluates classification accuracy, null accuracy, confusion matrix, false positive rate, precision of positive value, and AUC. I

implemented the classification accuracy and classification error to tell how often the classifier was correct or incorrect. After, I ran eight different machine learning techniques on the data, however I realized that eight methods was too many for me to analyze in the time allotted for this research. Thus, the following three machine learning methods were utilized on data: Random Forest, Decision Tree and Logistic Regression. I decided to use different methods to see which technique gave the best results. I then could compare to see which method had the highest success rating.

4.2 Methods

As stated previously, the machine learning methods used were: Random Forest, Decision Tree and Logistic Regression. Random forests are an ensemble learning method for classification by constructing a multitude of decision trees at training time. A decision tree are used to identify a strategy most likely to reach a goal. In this paper, our goal is to see if an individual sought treatment or not based on features involved in the technique. Logistic Regression is used to model the probability of a certain class or event. In this case, the probability of whether or not the person sought treatment based on the features. There were five different groups of features used.

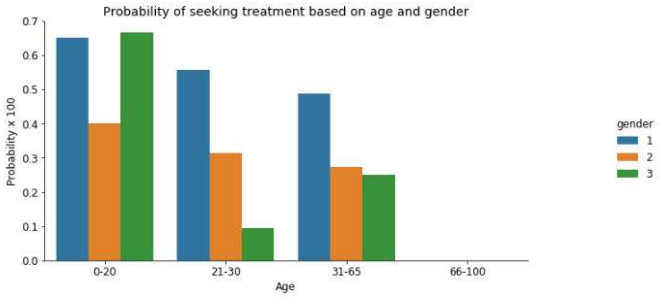
4.3 Feature Groups

In the Results section, I will be talking about the differences by groups. The first group involved the following nine features: age, gender, country, family history, mental health benefits at work, knowing mental health options under employer coverage, formal discussion about mental health wellness campaigns, protection of anonymity when discussing mental health condition, and ability to request mental health leave. The second group involved the following four features: mental health benefits at work, knowing mental health options under employer coverage, protection of anonymity when discussing mental health condition, and ability to request mental health leave. The third group involved the following nine features: if consequence is given by talking about mental health, if consequence is given by talking about physical health, comfortability with talking to coworkers about mental health, comfortability with talking to coworkers about physical health, mental health taken as seriously as physical health, mental health benefits at work, knowing mental health options under employer coverage, protection of anonymity when discussing mental health condition, and ability to request mental health leave. The fourth group included eleven different features: age, gender, family history, comfortability with talking to coworkers about mental health, comfortability with talking to coworkers about physical health, mental health taken as seriously as physical health, mental health benefits at work, knowing mental health options under employer coverage, protection of anonymity when discussing mental health condition, and ability to request mental health leave. The fifth group includes all 23 features from the dataset.

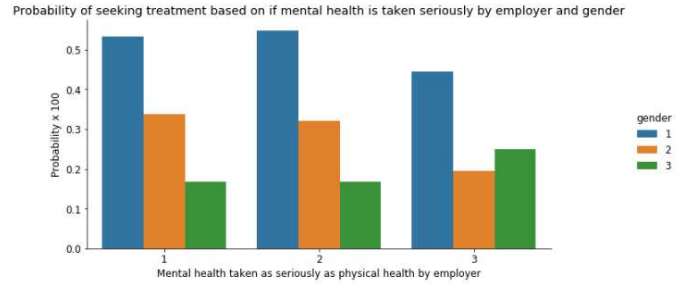
5. RESULTS

5.1 Bar graphs

Before performing the machine learning techniques, I decided to graph the data to get a sense of it. The first graph, shown below, dealt with probability of seeking treatment based on age and gender.

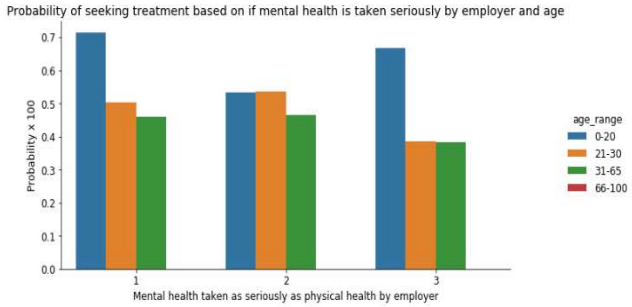


Genders include male, female, and other and are ranked 1,2 and 3. Ages were grouped by 0-20, 21-30 and 32-65. The results concluded that 0-20 year old individuals that classified themselves as “Other” gender were the most to seek treatment for their mental health with close to 70% probability. This makes sense because individuals classifying as “Other” have a more difficult time with acceptance in the world which causes poor mental health. Males were second highest to seek treatment. A bias in this study is that this data was collected in the tech industry. Primarily, men are the majority in the tech industry over women. Thus the data collected was mostly male responses. This remains true for the rest of the bar graphs. The probability of seeking treatment based on whether mental health is taken as seriously as physical health in the workplace was a huge element that I wanted to results to. The graph shows the probability of seeking treatment based on employers take on if mental health is as important as physical health differencing by gender.



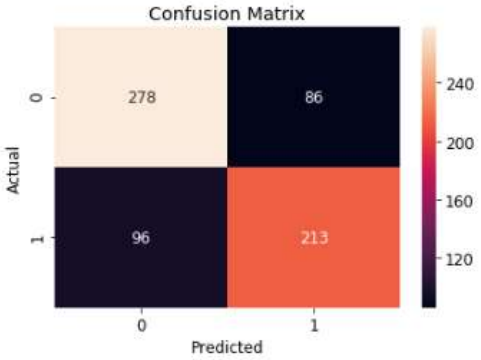
The x-axis 1, 2, 3 map to Yes, I don’t know and No to an employer taking mental health as serious as physical health. The results are heavily weighted for males. This is due to the tech industry bias. The next graph shown measures the probability of seeking treatment based on employers take on if mental health is

as important as physical health differencing by age.



5.2 Accuracy Scores

All accuracy scores are over 60%. This means that the results are influenced. Overall, the best accuracy results were from groups 1 & 4 for all machine learning methods except random forest. The best accuracy results were from groups 1 and 5 for the random forest method. Beginning with logistic regression method, group number 4 had the highest accuracy score with 72.96%. This means that over 72% of the time the model predicted correctly. The precision score for group 4 measured at 71.24%. This number represents how accurate the model is out of those predicted positive and how many of them were actually positive.



For example as shown above by the confusion matrix, the true positive for group 4 was 213/673 total predictions. The false positive was 86/673 total. Adding both of those and dividing the true positive number by that total gives us the precision score. Group number 1 had the second best accuracy with a score of 72.07%. Both groups (1 & 4) included age, gender, and family history as features. These three features were not included in groups 2 & 3. The dropping of the previous features in groups 2 & 3 give substantial reason why the accuracy scores for groups 1 & 4 were more than 10% higher. Decision tree was the next technique used on the dataset. Similar to logistic regression, groups 1 & 4 had the highest accuracy score. Groups 1 & 4 performed the same, with an accuracy score of 70.28%. The random forest method had similar results to both the logistic regression and decision tree methods. However, group 5 which included all 23 features ranked second best in accuracy score compared to group 1 Group 1 had predicted correctly 70.28% of the time. The random forest model's precision score for group 1 measured at 69.75%.

5.3 Area Under the Curve

Area under the curve or AUC, ranges in values from 0 to 1. A model whose predictions are 100% wrong has an AUC of 0.0. One whose predictions are 100% correct has an AUC of 1.0. This machine learning measurement evaluates how well predictions are ranked and also measures the quality of the model's predictions irrespective of what classification threshold is chosen. For logistic regression, the highest AUC score as from group with .73. This means there is a 73% chance that the model would be able to distinguish between positive and negative class. For the decision tree method, the highest AUC measurement was from group 4 with .71. Finally, group number 5 had the highest AUC score for the random forest method with a score of .71. Similar to the accuracy scores, the logistic regression model has the best AUC.

6. DISCUSSION

After analyzing the results of each algorithm, the decision tree method, specifically group 3, did not perform as well as logistic regression and random forest. A reason for the decision tree not performing as well is the case of overfitting. Decision trees are vulnerable to become biased to the classes that have a majority in the dataset. In this dataset, the majority of the results were from males, hence the bias and poor execution.

The best performance was accomplished by the logistic regression model. When doing research on previous projects and relevant work dealing with mental health, the most popular approach was logistic regression. Overall, the highest accuracy score was from group 4 using logistic regression method with a score of 72.96%. The highest AUC score, overall, was also from group 4 using logistic regression. The groups with features age, gender, and family history outperformed the groups that did not include those 3 features. As previously discussed, one downfall of this project was the dataset not having an equal proportion of responses from males versus females in the training dataset. Another downfall that this project has is the amount of unknown answers. This can be due to employees not paying attention to mental health because it does not affect them or it could be due to employers not starting the conversation with their workers. As you can tell from the bar graphs above, the "Not Sure" answers (represented by the number 2) are just as high as the "Yes" and "No" answers.

7. CONCLUSION

In this project, I attempted to classify whether or not an individual sought treatment based on the features surrounding. The key findings show us that the best classifier to use on this dataset is the logistic regression with the features from group 4. That method gave us close to 73% accuracy score. The decision tree model did not perform as well as expected with an accuracy score as low as 59% from group 2. These results, although higher than a 50/50 probability, show us that there is still room for improvement when it comes to determining whether or not someone sought treatment or in the future should seek treatment. The results also show that the features used for each machine

learning model plays a significant role in how accurate predictions of the models are.

The growth and execution of a workplace mental health guideline would benefit the health of employees, both personally and professionally, as well as the business overall. An employer's business would be benefited by a mental health policy because the productivity of the employees for the company would contribute to the well-being of the business as a whole. Mental health should be treated as any other illness or health concern. An individual who receives help at work will improve functioning at work which leads to financial stability. However, taking care of an individual's mental health before it turns into a mental illness is the best outcome anyone could ask for. As an employer, find a team of trusted and innovative mental health care providers to offer to employees. This way, individuals with mental health conditions persevere at work rather than in an lose hope in an institution. In conclusion, individuals who need help seeking treatment or have sought treatment for a mental health condition should not feel ashamed to do so.

8. FUTURE WORK

Had there been more time to clean and analyze the data, I would have figured out a way to combine all years together beginning with 2014 to 2019. This way, the responses between male and female would be less unbalanced since more women since 2014 have been entering into the tech industry. As more women start to enter into the tech industry, I have little doubt that responses from women became higher than responses from men in the latter years of data collection by OSMI. With more unbiased data, the accuracy scores would be far better off. Lastly, I would have liked to complete the analysis on the Deep Neural Network method with this dataset. Since mental health is the psychological state of an individual, the technology and performance behind DNN would be an interesting comparison to the techniques already used in this paper.

9. ACKNOWLDGEMENTS

Thank you to everyone who has been by my side through my years at Mercyhurst University – I did it!

10. REFERENCES

[1] "What Is Mental Health and Mental Illness?" *Workplace Mental Health Promotion*, 2017, wmhp.cmhaontario.ca/workplace-mental-health-core-concepts-issues/what-is-mental-health-and-mental-illness.

[2] Motsiff, Dawn. "Mental Health In The Workplace: 5 Things You Need To Know." *Insperity*, 29 Jan. 2020, www.insperity.com/blog/mental-health-in-the-workplace/

[3] Harvard Health Publishing. "Mental Health Problems in the Workplace." *Harvard Health*, www.health.harvard.edu/newsletter_article/mental-health-problems-in-the-workplace.

[4] Vargas, Susan. "'It's Not an Easy Conversation': Mental Health in the Workplace." *Safety+Health Magazine*, Safety+Health Magazine, 13 Dec. 2019, www.safetyandhealthmagazine.com/articles/17489-its-not-an-easy-conversation-mental-health-in-the-workplace.

[5] "Kate Spade Suffered From Anxiety and Depression, But No Warning Signs of Suicide, Husband Says." *MedicineNet*, MedicineNet, 7 June 2018, www.medicinenet.com/script/main/art.asp?articlekey=212862.

[6] "Mental Health in the Workplace." *World Health Organization*, World Health Organization, 9 Aug. 2019, www.who.int/mental_health/in_the_workplace/en/.

[7] Sime, Carley. "The Cost Of Ignoring Mental Health In The Workplace." *Forbes*, Forbes Magazine, 17 Apr. 2019, www.forbes.com/sites/carleysime/2019/04/17/the-cost-of-ignoring-mental-health-in-the-workplace/#a7b664c3726a.

[8] Maulik, Pallab K. "Workplace Stress: A Neglected Aspect of Mental Health Wellbeing." *The Indian Journal of Medical Research*, Medknow Publications & Media Pvt Ltd, Oct. 2017, www.ncbi.nlm.nih.gov/pmc/articles/PMC5819024/.

[9] Rajgopal, T. "Mental Well-Being at the Workplace." *Indian Journal of Occupational and Environmental Medicine*, Medknow Publications, Sept. 2010, www.ncbi.nlm.nih.gov/pmc/articles/PMC3062016/.

[10] Institute for Health. "Mental Health in the Workplace: A Call to Action... : Journal of Occupational and Environmental Medicine." *LWW*,journals.lww.com/joem/FullText/2018/04000/Mental_Health_in_the_Workplace__A_Call_to_Action.5.aspx

11. Data Sources

[1] Open Sourcing Mental Illness, LTD. "OSMI Mental Health in Tech Survey 2016." *Kaggle*, 17 Nov. 2019, www.kaggle.com/osmi/mental-health-in-tech-2016.

[2] Open Sourcing Mental Illness, LTD. "Mental Health in Tech Survey." *Kaggle*, 3 Nov. 2016, www.kaggle.com/osmi/mental-health-in-tech-survey.

About the authors:

Tiffany Gonzalez is a 2nd year Data Science Graduate Student at Mercyhurst University. Tiffany has an undergraduate degree in Accounting. Tiffany works at Erie Insurance as a Software Engineer. She has a strong passion for mental health and advocates for mental illness.

Differentiating Benign from Malicious Portable Executables with Machine Learning

Griffin Noon
Computing & Information Science
Mercyhurst University Erie, PA
gnoon86@lakers.mercyhurst.edu

Peter Chuzie
Computing & Information Science
Mercyhurst University Erie, PA
pchuzi50@lakers.mercyhurst.edu

Hasanain Alsaeedi
Computing & Information Science
Mercyhurst University Erie, PA
halsae54@lakers.mercyhurst.edu

Sebastian Pardo
Computing & Information Science
Mercyhurst University Erie, PA
spardo12@lakers.mercyhurst.edu

Zach Kozlin
Computing & Information Science
Mercyhurst University Erie, PA
zkozli13@lakers.mercyhurst.edu

ABSTRACT

A constant flow of new and evolving malware makes it nearly impossible for traditional anti-virus systems to mitigate these threats. A major issue that exists for the current approaches, such as signature-based detection, is that current malware is using polymorphic and metamorphic techniques. Polymorphic and metamorphic techniques allow for code to change characteristics about itself so that it will no longer match its previous strand, but still accomplish the same malicious tasks. This makes it difficult for signature-based detection to recognize the malware. Therefore, it is necessary to create a system that looks for certain characteristics or actions that are often found in association with malware, and that ultimately does not rely solely on previously defined stands. This can be accomplished with a machine learning model. Through using machine learning the detection software will no longer depend on depend on large databases of known malware strands, but use the features of a portable executable to make a decision on whether the file is malicious or benign.

KEYWORDS

Target, Features, Accuracy, Malicious, Benign, DecisionTreeClassifier, RandomForestClassifier

1 Introduction

Computers help in everyday life by increase productivity, have access to all our information, as well as many other things so it is vital to keep them running smoothly. However, computing devices are vulnerable to several types of malware that are difficult to detect and that can hinder functionality. Malware can be defined as a software that is specifically designed to disrupt, damage or gain unauthorized access to a computer system. Common types of malware are worms, adware, trojan horses and spyware. The current approach to handling malware is by relying on the use of known malware data signatures to identify a piece of malicious code that may be attempting to infiltrate your system. The problem with this approach is that new malware is constantly being created with undetectable malicious signatures. Due to this, it is our goal to teach a computing system to be able to differentiate between a benign and malicious file without relying on any explicit signature database. This can be done by developing a machine learning model. Machine learning provides systems the ability to automatically learn from experience without being additionally programmed. Hence, it is our objective to train our machine learning model to distinguish between benign and malicious files by learning their characteristics/features from large sets of data.

2 Relevant Work

Detecting malware with machine learning is a concept that is continually being improved on. The following are some examples of research that is similar to that of this paper. Historically, there have been several attempts at creating an effective PE malware detection system. In 2009, a researcher extracted 189 features from

PE file segments and used algorithms like Principle Component Analysis to select the most relevant features with a 99% success rate. This project was re-examined and a group concluded that 9 features are needed to achieve the same accuracy which were generated with an external software, so the overall performance relied on that software. The next major step in detecting malicious PEs was utilizing static and dynamic features which included "Op-code n-grams, API n-grams and embedded behavior graphs;" this method obtained a 95% accuracy rate with a notable false positive rate of 2%. One other attempt was by Salaginov and it compared Neuro-Fuzzy and Artificial Neural Network on detecting ten malware families and ten categories. This work relied on an external software and boasted a significantly lower success rate of 81% and a drastically higher false positive number at 20%. One gap that is noted in research is that there has been little to no investigation done using a small number of static features that were directly extracted from the PE files, though. Moving forward there were several attempts at creating detection models with ideas from one or several groups' previous research with varying success rates.

3 Understanding Portable Executable Features

Before creating models it is important to understand the basic structure of portable executables (PE) and their features. It will help us understand why we selected the feature we did. In addition, understanding these properties of PE file will help us understand how malicious files differ from benign files. The PE file format is executables used in 32-bit and 64-bit versions of the MS Windows operating systems (OS).[1] It is a file structure that holds the information needed for the Windows OS loader to manage the executable code wrapped inside. The basic structure of a PE consists of two sections, which can be further subdivided into smaller subsections. The two main parts are the Header and Sections as seen in Figure 1.[2]

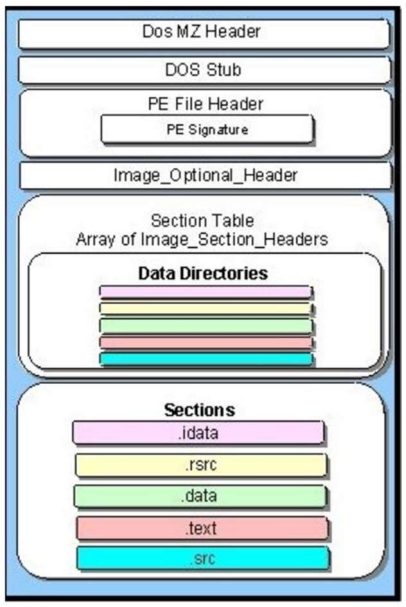


Figure 31: Structure of a PE file.

All PE files start with the Dos header interface, which occupies the first 64 bytes of the file. This allows for DOS to recognize the file as a valid executable so that it can be run in DOS stub mode.[3] As can be seen in figure 1 there is a list of structures that come under the DOS header. Two of the most important structures are 'magic' and 'ifanew'. The field 'magic' or 'e_magic' is also known as the magic number. It identifies an MS-DOS compatible file. If it is MS-DOS compatible then the magic number will be set to 0x54AD, which is the ASCII characters for MZ.[4] Next, the 'ifanew' field is a required element in the DOS Header because it turns the EXE file into a PE. This allows the windows loader to skip the DOS stub and go directly to the PE header. It does this because the first few hundred bytes of a PE file are normally taken by the MS-DOS stub, but the 'ifanew' field always for it to skip directly to the PE File Header. As seen in figure 1 the PE File Header contains information on what the rest of the file looks like. It includes information such as the location, signature, machines, Number Of Sections, Size Of Optional Header, and Characteristics.[5] The signature specifies the intended target OS, and is the first word of the file header. It also is used as an indicator for a file to determine if it is a known trusted file, untrusted or unknown.[6] Next, Number of Sections details the size of the section table which follows the header (as seen in figure 1). The Size Of Optional Header details the size of the optional header that is needed for the executable file. The Characteristics attribute indicates what the file is, such as a DLL that has a flag call of Image_File_dll. Next, we have the Image_Optional_Header and this contains

important information about the image. It includes the initial stack size, program entry point location, preferred base address, OS version, section alignment information, among others. One of the most important parts of the Image_Optional_Header for this project is the entry point location which indicates the location of the entry point for the application. Some other important features that fall within the optional header are configuration_table, debug, import table, export table, and exceptions table. These are important to take note of because they assist in determining if a file is malicious or benign. If debugging or configuration_table are active or not can point to a file being malicious or benign. Moreover, there are many standard fields in this section, but they are outside our scope.

The second part of a PE is called Section Table. It immediately follows the optional header, and this portion contains the main content of the file. This includes code, data, resources and other executable files. Normally within the Section it contains nine predefined sections (not all are shown in figure 1). We will cover some of the more important such as .TEXT which contains all the code segments. .BSS represents the uninitialized data for the application. .RDATA shows the read-only on the file system. .RSRC has info on resources of a module, such as icons an images that are part of the file's resources. The .EDATA section possesses the export directory for an application , if present it holds names and addresses for export functions. Next, the .IDATA section contains information about the import functions such as directory and import address table. The .PDATA section is responsible for exception handling in the file. Another important part of Sections is the Thread Local Storage (TLS). This allows for processes in a PE to be threaded, so each time a process creates a thread a TLS is build and the thread uses the '.tls' as a template.[7] This is an important feature because it is one that is often abused in malware creation. For example, since TLS callbacks run before the debugger can gain control, it can make changes that result in it being impossible for the file to be debugged by ordinary means.[8] This is just one example of many ways the tls can be used in a malicious ways since it runs even before the main entrypoint of the PE. Overall, this is a lower level over view of all the details that go into a PE, but this covers some of the most important features that are used later in our model to determine if a PE file is malicious or benign.

2 Proposed Solution

Now that we understand the basic structure of a PE we can begin exploring the solution for creating a machine learning model that can differentiate between malicious and benign PE files. To begin, a dataset of 200,000 samples with 437 features related to PE files was collected and available for our use by VirusTotal. This dataset was half malicious files and half benign allowing us to approach our problem using supervised machine learning. It was also determined that our problem was a classification type because there were only two values to predict, malicious or benign. In order to make the problem processible by a model a value of one was assigned to malicious files while zero meant benign. In order to reduce the number of features being used data exploration or feature selection was done by finding the correlation between our features and the target. The algorithm seen in Equation 1 was used to accomplish this.

$$COV(X,Y) = \frac{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right)}{n-1}$$

Equation 1: Correlation equation.

Sost features got a correlation of around 0% which is displayed in the heatmap in Annex A. Therefore, it was decided to only use features which had a negative or positive correlation of 10% of higher, which resulted in 46 features as depicted in Annex B. Among these some of our most important features included if a file has a signature, debug, configuration, export, import, and TLS which are all mentioned in the previous section. Another two important features are two ratios: the code section ratio and the pec section ratio. This is used to analyze the PE sections containing code or data. For each of these it will help to determine the average of their size vs their virtual size and will inform the model if and how the PE is packed, compressed, and obfuscated.[9] Next, the feature "has_exceptions" is also highly correlated with the target, and this information is found in 'pdata' within the Sections portion of a PE. Also, the feature 'has_nx' which determines if the PE has the NX (no-execute) bit is highly correlated. The NX bit is used to segregate memory for use to either store code or data. It is also used to stop certain malicious software from taking over computers by inserting their code into another program's data storage area and running their code in the other programs section, similar to the buffer overflow concept.[10] Ultimately, the 46 features with a 10% correlation or higher are what will be used to train the model.

3 Evaluation & Discussion

Before testing or deciding on a model. The data was divided into training and testing sets. The training set made up 80% of the data and the other 20% was reserved for the testing set. Once this was determined, the base model was run in the

form of a DecisionTreeClassifier. The decision tree had a max depth of two and the criterion was set to entropy (Annex C). The first split of the model was determined by the 'code_sections_ratio' because it was highly correlated with our target and was based on 159,975 samples. If the 'code_sections_ratio' was less than or equal to 0.392 and the class was 1 it moved down the tree toward false or malicious. When the 'code_sections_ratio' is above 0.392 the file moved toward benign. On the second level, if the file was moved toward benign then the next feature is 'pec_sections_ratio'. If the 'pec_sections_ratio' is less than or equal to 0.321 the file is classified as benign, but if it is higher then it is classified as malicious. If after the first level the file moved toward malicious the next feature is 'has_signature'. When 'has_signature' is less than or equal to 0.5 the file is classified as malicious and if it is less it is considered benign. This base model managed to achieve an accuracy of 75%. This was impressive for the base model, but obviously had room for improvement. To improve the accuracy we continued to do model tuning by increasing the depth and switching the parameters of the criteria from gini to entropy. The next model had a depth of three and a criterion of entropy and resulted in 79% accuracy. This process continued until we reached a max-depth of 12 and a criteria of gini which provided us with our greatest accuracy using the DecisionTreeClassifierModel. Unfortunately, an image of this tree could not be added to the document due to its size.

4 Results & Discussion

With a max depth of 12 and criterion of gini our DecisionTreeClassifier model was able to achieve a 96% accuracy. However, to further evaluate if the accuracy could improve a different model was applied. The RandomForestClassifier was selected to see if it could improve upon the 96% accuracy achieved with the DecisionTreeClassifier. The RandomForestClassifier is a collection of decision trees run in parallel and was run using the 'fill' method (as seen in Annex D along with the parameters used). The reason this was used is because a combination of learning models will increase the overall score. For our RandomForestClassifier model, 100 DecisionTrees were used. After running this model, it resulted in an improvement of 2% from the DecisionTreeClassifier and achieved an accuracy of 98%. At this point we stopped attempting to improve the model because the results provided a higher accuracy.

Therefore, the most accurate model to determine malicious from benign PE files is the RandomForestClassifier model with 100 decision trees. This was determined by testing the accuracy of different models and tuning them to their most accurate results, and then comparing them.

5 Conclusion

Through this process we learned the importance of model tuning, as well as being willing to approach the problem with different models. We were also able to determine that a new approach to malware detection was needed because signature and behavior detection is unable to effectively protect against new advanced threats. Therefore, the need for a malware detection system that is integrated with machine learning will help to improve the security of systems as well as cyber security as a whole.

REFERENCES

[1] Patricia S. Abril and Robert Plant, 2007. The patent holder's dilemma: Buy, sell, or troll? *Commun. ACM* 50, 1 (Jan, 2007), 36-44. DOI: https://doi.org/10.1145/1188913.1188915.

[2] Sten Andler. 1979. Predicate path expressions. In *Proceedings of the 6th. ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL '79)*. ACM Press, New York, NY, 226-236. DOI:https://doi.org/10.1145/567752.567774

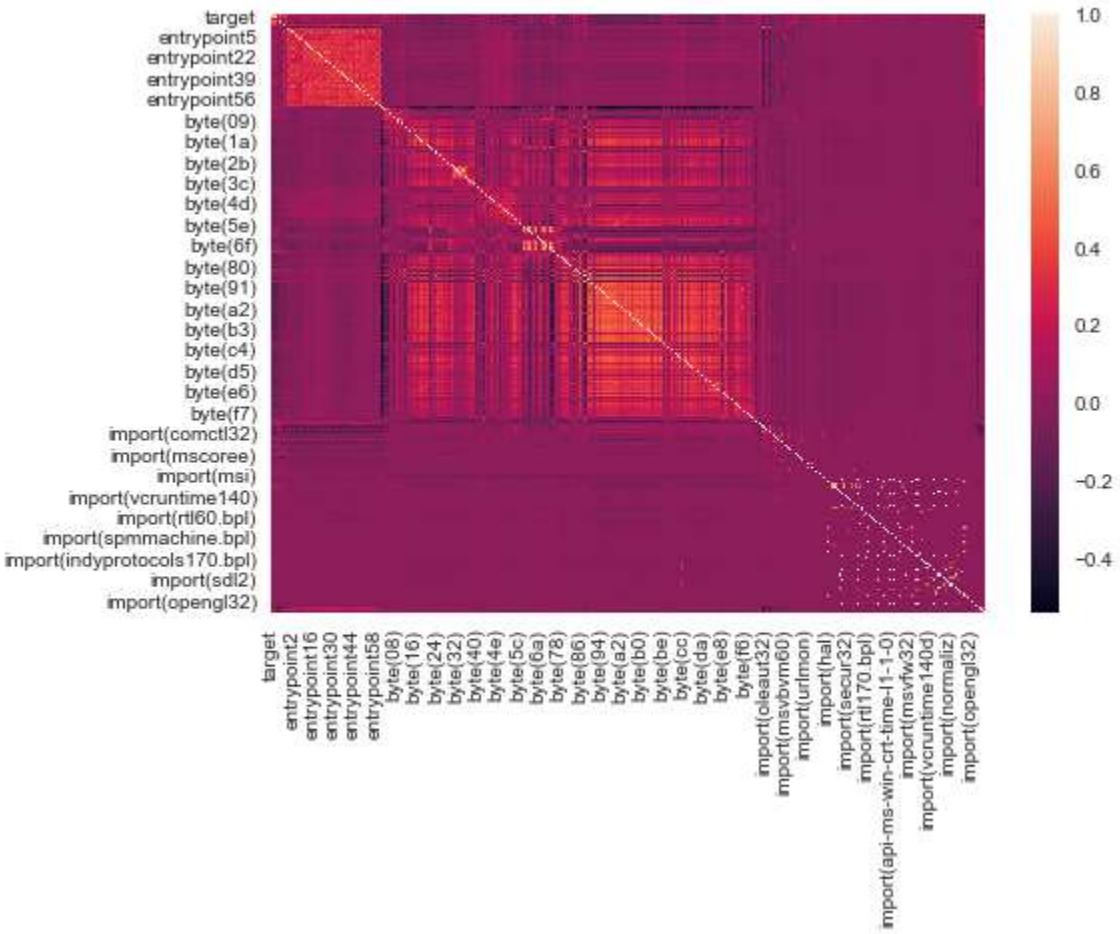
[3] Ian Editor (Ed.). 2007. *The title of book one* (1st. ed.). The name of the series one, Vol. 9. University of Chicago Press, Chicago. DOI:https://doi.org/10.1007/3-540-09237-4.

[4] David Kosiur. 2001. *Understanding Policy-Based Networking* (2nd. ed.). Wiley, New York, NY.

Annexes

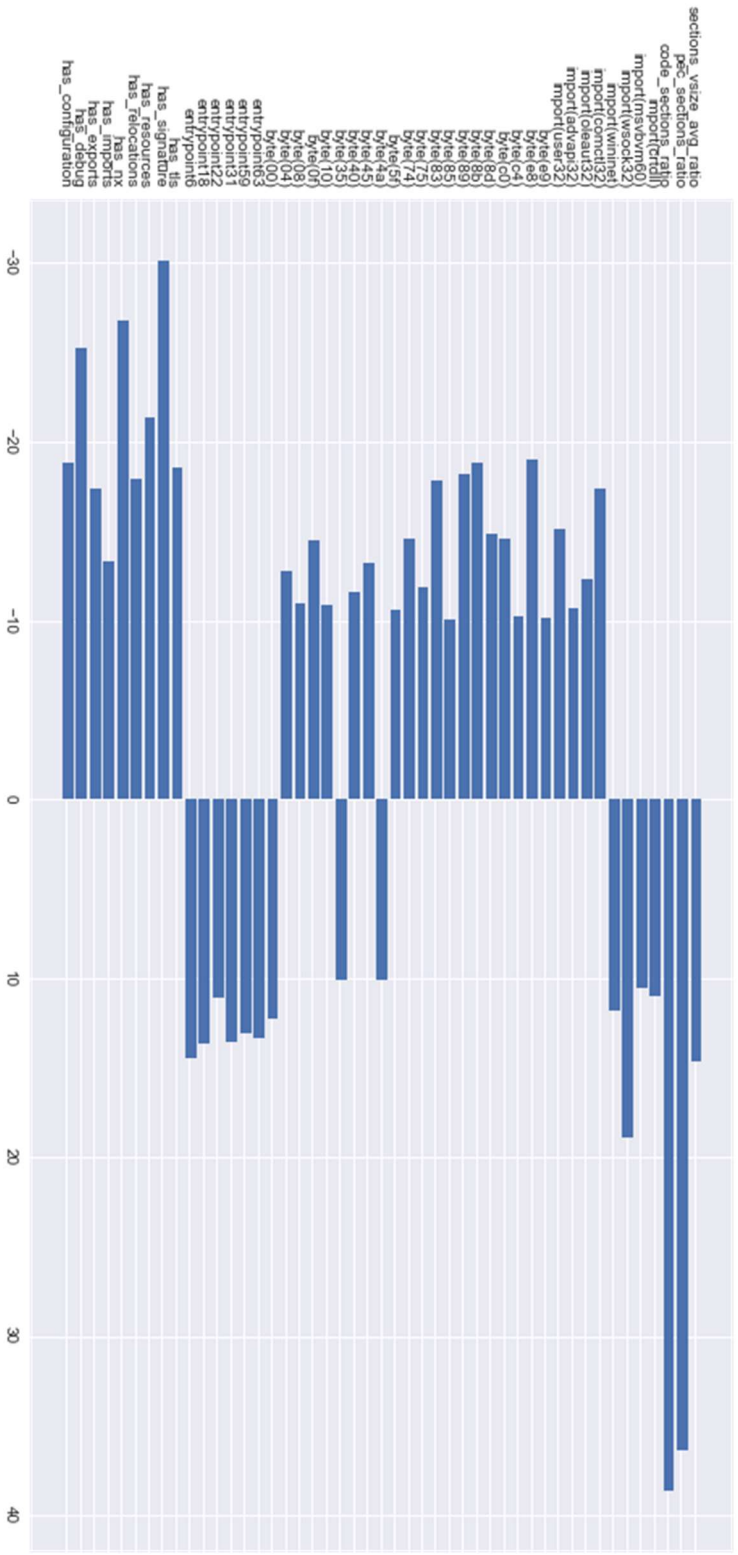
Annex A:

A heat map depicting the correlation of features to the target (malicious or benign files).



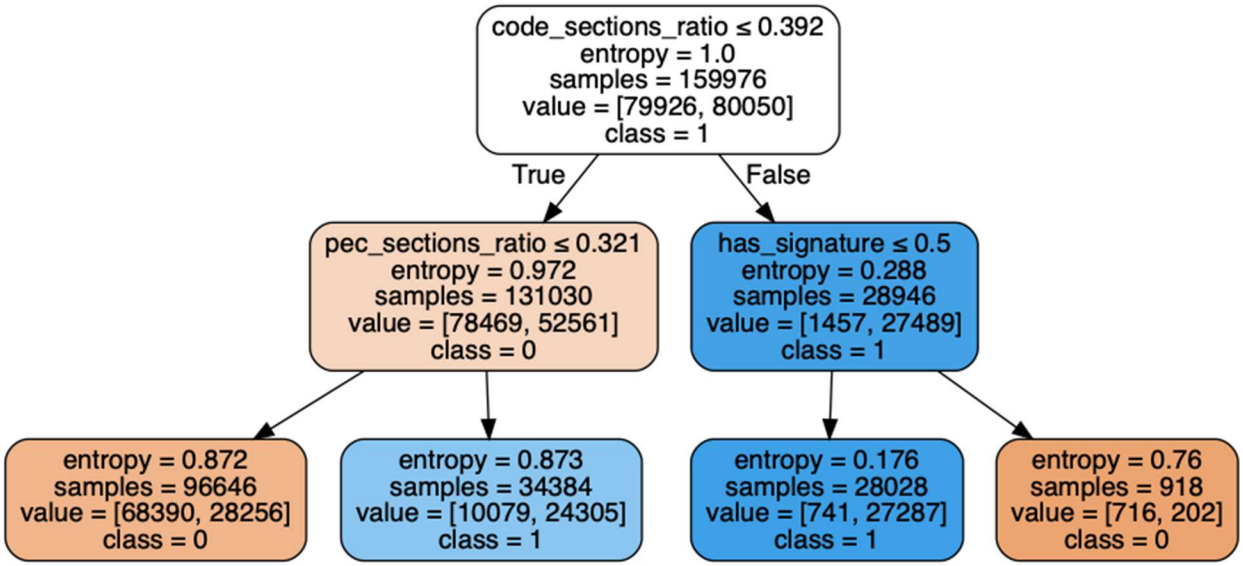
Annex B:

A chart depicting the top 46 features with a correlation greater than or equal to the target.

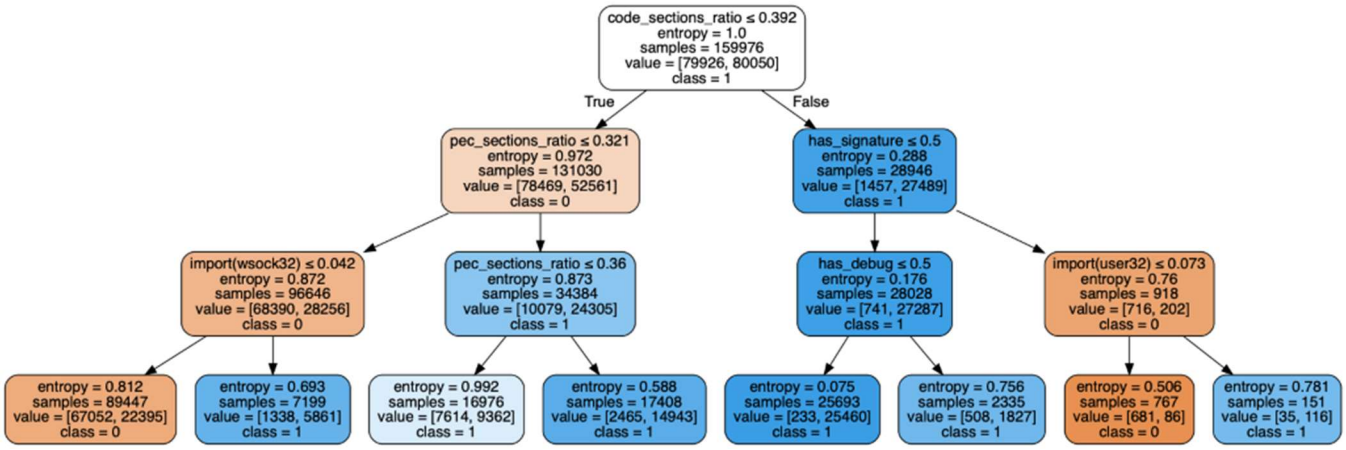


Annex C:

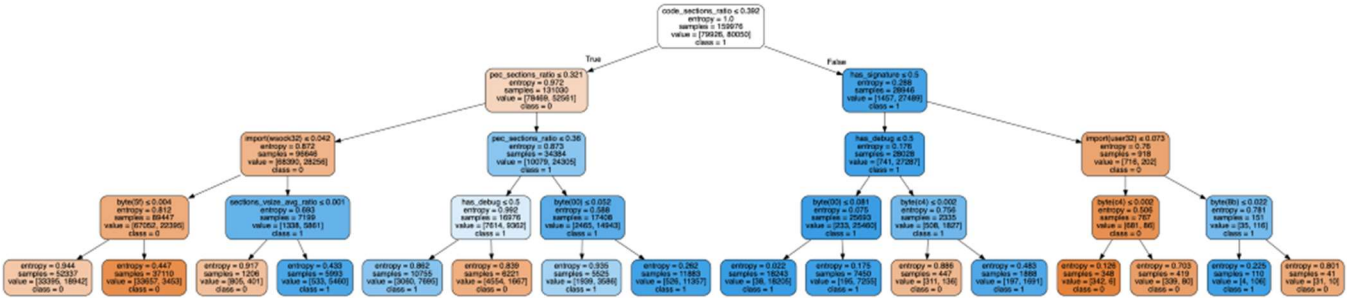
This is a visual depiction of the base decision tree model that resulted in 75% accuracy.



The following is a visual depiction of the decision tree with a max-depth of 3, resulting in an accuracy of 78%.

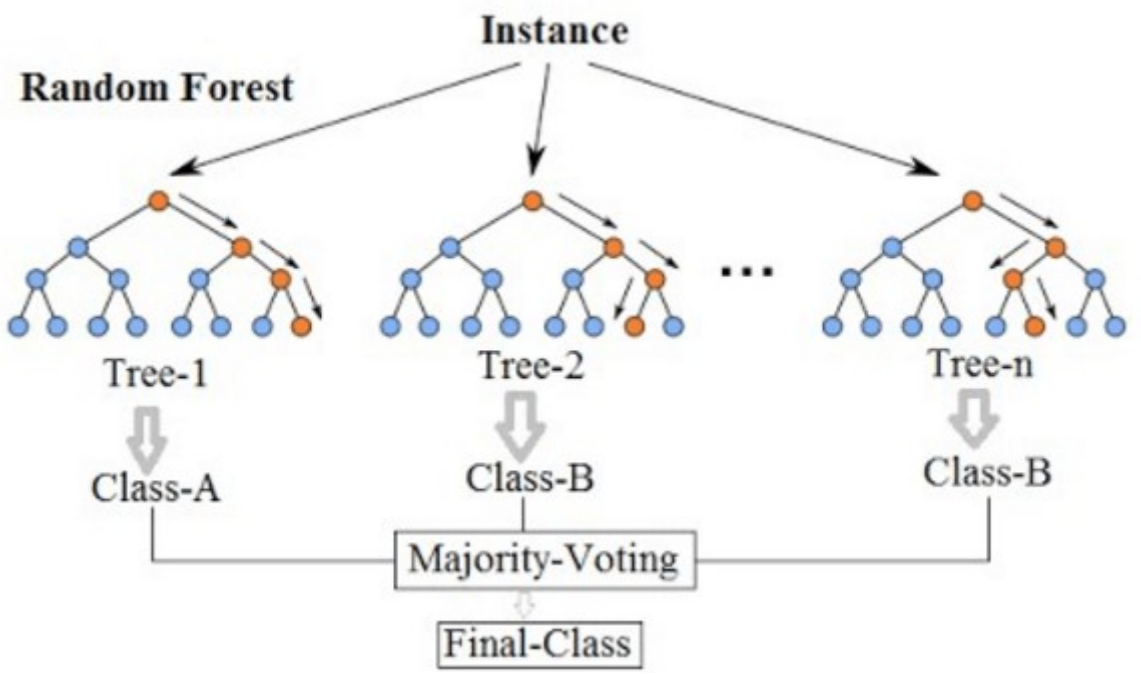


The following is a visual depiction of the decision tree with a max depth of 4, resulting in an accuracy of 80%.



Annex D:

The following depicts the basic structure of a random forest as well as the parameters used.




```
                RandomForestClassifier(bootstrap=True,
                                       ccp_alpha=0.0,
                                       class_weight=None,
                                       criterion='gini',
                                       max_depth=None,
                                       max_features='auto',
                                       max_leaf_nodes=None,
                                       max_samples=None,
                                       min_impurity_decrease=0.0,
                                       min_impurity_split=None,
                                       min_samples_leaf=1,
                                       min_samples_split=2,
                                       min_weight_fraction_leaf=0.0,
                                       n_estimators=100,
                                       n_jobs=-1, oob_score=False,
                                       random_state=None,
                                       verbose=0,
                                       warm_start=False))],
flatten_transform=True, n_jobs=-1, voting='hard',
weights=None)
```

[1] https://resources.infosecinstitute.com/malware-researchers-handbook/#article
[2] https://resources.infosecinstitute.com/malware-researchers-handbook/#article
[3] https://resources.infosecinstitute.com/2-malware-researchers-handbook-demystifying-pe-file/#gref
[4] ibid
[5] https://resources.infosecinstitute.com/2-malware-researchers-handbook-demystifying-pe-file/#gref
[6] https://blog.kowalczyk.info/articles/pefileformat.html
[7] https://resources.infosecinstitute.com/2-malware-researchers-handbook-demystifying-pe-file/#gref
[8] https://web.archive.org/web/20100626014510/http://blogs.technet.com/b/mmpc/archive/2010/06/21/further-unexpected-resutls-sic.aspx
[9] https://www.evilsocket.net/2019/05/22/How-to-create-a-Malware-detection-system-with-Machine-Learning/
[10] http://h10032.www1.hp.com/ctg/Manual/c00387685.pdf


1926
MERCYHURST
UNIVERSITY