Automated Classification Methods of Darknet Market Listings:   
a Text-Based Analysis

I. Introduction

Motivation:  
 Beyond the extent of what can be found on the internet through a regular browser lies the Dark Web: an online platform that operates on private networks and requires special software, configurations, and authorizations to access. These barriers to entry and the clandestine nature of the Dark Web are leveraged by users to operate in anonymity, which has popularized Darknet markets. Here, escrow services are provided that facilitate transactions involving drugs, cyber-arms, weapons, counterfeit currency, stolen credit card details,forged documents, unlicensed pharmaceuticals, steroids,and other illicit goods, and are characterized by cryptocurrencies as payment and vendor feedback systems similar to that of eBay. As more users have increased access to anonymizing networks such as the Tor network, the growth of illegal activities in the Dark Web continues to grow. When dealing with this growth of illegal activities, legislation, technical abilities, and data are vital components for strategy formulation, but as it stands researchers and law enforcement bodies in many countries are not adequately equipped to deal effectively with the illegal activities on the Dark Web, especially when collecting data. Oftentimes, the methods employed depend upon error prone manual techniques (Kinningham, 2015). As such, a fundamental step to addressing the issue first requires a general understanding as well as the acquisition of data on product listings which can help authorities aggregate data. In this paper, we propose an approach that can standardize text data and appropriately classify it in a manageable, interpretable framework.

Purpose:   
 By acquiring data on Darknet listings, we begin the next step of creating a method for proper and efficient classification of the data. We constructed a machine learning algorithm to classify product listings posted to anonymous marketplaces as being either narcotics related or some other listing. This choice of classification would provide a method that could more efficiently separate listings more obviously narcotics related, and thus illegal, and those which are not. As mentioned, law enforcement officials and interested researchers usually have relied on an expert to manually tag each listing to properly categorize each product sold. Overall, this approach is often time consuming and inefficient considering the sheer number of listings available on Darknet markets in addition to the unrelated legal listings included in these databases as well. The constructed algorithm seeks to automatically categorize listings with high accuracy, using the listing’s text included in the product title and description, to provide those interested parties with an efficient method for classification of these types of databases.

Quick overview of methodology:   
 After generally optimizing and cleaning our data, we employed various methods to construct our algorithm. First, we used the Bag of Words approach and Term Frequency-Inverse Document Frequency (TF-IDF) to extract our word features. Extracting these word features will allow us to cut the data set down and leave us with a relevant dataframe. Next, we partitioned our data into a training set and a testing set. We can use different methods of classification on these training sets to test which method could yield the highest accuracy in classifying this data. One method used was the Linear Discriminant Analysis method, a reduction technique where each successive transformation maximizes the between-class variance relative to the within-class variance and transforms the feature space to a linear space. Another method was the Naïve Bayes Classifier, a classification method used to calculate conditional probability using Bayes theorem. The final method of classification used was the feed forward neural network, which is a deep supervised learning model that learns the value of the parameters that result in the best function approximation. We employed the grid-search approach to minimize overfitting for the classification methods. Once the grid-search approach was implemented, specific tuning methods were performed on both the Naive Bayes and feed-forward classification techniques. This was done through a Laplace correction sequencing on the Naive Bayes method and a network decay sequencing on the feed-forward neural network. Once the models’ parameters were tuned and the models were cross validated we observed the results to determine which model provided the most accurate and efficient method for classifying this dataset.

II. Related Works

Kinningham:   
 We have found one academic paper that has attempted to create an automatic product categorization process for listings on anonymous markets. In their paper, the researchers developed a machine learning algorithm combining unsupervised feature selection with a supervised classifier that classify product listings according to the type of product being sold. They extracted word features using Term Frequency-Inverse Document Frequency (TF-IDF) for feature extraction, employed Singular Value Decomposition (SVD) for feature selection, and Support Vector Machines (SVM) for final classification. Finally, they used cross-validation to select hyperparameters and tune their algorithm. In comparison to other baseline models, their algorithm yielded a higher accuracy, in part due to the algorithm’s ability to take advantage of the structure of a large unlabeled data set through their methods. For our project, we will perform a similar process but with a much smaller set of labeled data and using different classification models that can later be applied to unlabeled datasets.

III. Data

Gwern (2014-2016):  
 Our dataset comes from a researcher by the name of Gwern, who compiled a comprehensive list of anonymous marketplace product listings. The data set, under the moniker “grams”, was compiled by scraping darknet marketplaces daily from June 6th, 2014 to July 7th, 2015. The data itself represents standardized product listings for a variety of goods and services offered by dark web vendors. The significance of this data cannot be understated - this data collection represents an unfiltered look into the workings of the dark web. Given the dubious nature of many darknet markets, enforcement agencies have since seized the domains for these sites. Scrutiny from these agencies has disrupted large, centralized marketplaces for sellers. As a result, data collection has become significantly harder for researchers to synthesize and compile. We attempt to address this issue in our paper via a sampling strategy that incorporates multiple, smaller samples from markets.

Sample strategy:  
 Our original dataset consists of all product listings contained in the grams dataset from two popular darknet markets: Agora and Evolution. Because this dataset contains all listings between 2014 and 2015, we sought to significantly reduce the number of features in the dataset. First, we randomly sample approximately 50% of the original one million observations. Subsequently, we remove repeated listings to avoid redundancy. Finally, we balanced the data to include equal listing categories. Figure One illustrates the final outcome of our balanced data by product listing category. After other text standardization techniques, our dataset includes 8,886 standardized product listings that will be input into three classification models. Figure Two provides some basic insight on the overall price distributions of our listings grouped by coded categories.

Figure Two

Figure One

Chart, histogram

Description automatically generatedChart, bar chart

Description automatically generated

Raw Data Features:  
 The grams dataset has the hash of the particular transaction, the seller, the price of the listing, the listing’s title, a subsequent description for the listing, the shipping options, and the date the listing was posted. Of these features, our analysis is centered around the title and description text data from individual listings. To better suit our model, we modified the dataset by creating a binary variable associated with each listing. This binary was coded (1) for narcotics related listings, and (0) otherwise.  In our analysis, we classify all drug-related product entries as narcotics related. As a broad category, this applies to psychoactives, hallucinogens, stimulants, sedatives, and research chemicals. Other listings are a larger, more amorphous class that encompasses both services and goods. Pornography, hacking services, and weapons were some of the most popular categories under our (0) category. We provide two sample listings that best illustrate the differences between the two listings below. Each listing has text under the title and description boxes. We use this information as the primary source in running our shrinkage and supervised learning algorithms for classification.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Market Name | Evolution | | Title | 100x GOLDBAR S 200MG MDMA | | Description | “100x Goldbars. This pills are made with very clean mdma and you will get a nice roll.” | | Price | 1.2 | | Category | 1 | | |  |  | | --- | --- | | Market Name | Agora | | Title | 10 Passport Scan of the U.K. | | Description | “10 Passport Scan in color and valid. Please indicate your male to female desired ratio.” | | Price | .3 | | Category | 0 | |

Figure Three: Sample Product Listings - Narcotics and Other

Training Data:  
 Before we partition our data into a training set and a test set, the text of each listing must be converted to features through the Bag of Words approach and TF-IDF. This allows us to convert our text data into numerical data which can be interpreted by our algorithm and models. These processes will be explained later in our methodology section. Once our data has been converted, we create a partition of our data. Around 75% of the data was then used to train our models.

IV. Methodology

Figure Four provides a brief overview of our data pipeline and classification methodology. After preparing our final dataset, we took steps to make our text data interpretable; we tokenize our product listings and then convert them into numeric values. After partitioning data into training and testing datasets, we train our classification models. Linear discriminant analysis does not have any parameters to tune, as it is a shrinkage method. In general, we utilize ten-fold cross validation to tune our hyperparameters for our naive bayes and feed-forward neural network models. After calculating the optimal hyperparameters, we run our models on the testing partition and provide confusion matrices detailing the accuracy of our models.

Diagram

Description automatically generated  
Figure Four: Condensed Data Pipeline

Bag of Words and N-Grams:  
 We must first begin with the bag-of-words approach, which is a way of representing text data as a vector when modeling text with machine learning algorithms (Stecallena, 2017). Essentially, each unique word in a phrase comprises a vocabulary which is used to form a string of numbers signifying the occurrence of each word in a phrase. To do this, we use the process of tokenization, which breaks up the text in the title and description boxes for each product listing into words, or tokens. We can sum up each unique token to create a count of how often each token appears, and we prioritize tokens which show up in at least 15 product listings. While tokenization and the bag of words approach will create a large feature size, it would allow us to capture the maximum amount of variations thereof. Furthermore, we remove all stop words - this ensures that common words with no meaning are not processed in our model. Next, we employ the bag of n-grams approach to improve information extraction. N-grams is a sequence of n-tokens that allows us to capture more context around each word surrounding each token as n increases (Boehmke, 2018). For instance, under 3-grams, the phrase “This pills are made with very clean mdma” can be converted to “This pills are”, “pills are made”, “are made with”, etc. For our purposes, we will use unigrams and bigrams to avoid a larger feature set.

TF-IDF:  
 We must convert our text data into numerical data; this is done by creating a term frequency-inverse document frequency (TF-IDF) matrix. Briefly, the TF-IDF matrix weights the importance of specific words (in our case, features) through two metrics: the frequency the word appears in a document next to a related category, and the inverse document frequency of the word across a set of documents. The term frequency of a word in a document is usually calculated by the raw count of instances a word appears in a document. In our case, the documents are the product titles and descriptions. The inverse document frequency of the word across a set of documents is the metric that measures how common or rare a word is in the entire document set. The closer it is to 0, the more common a word is. In combination, the algorithm for TF-IDF weights words by multiplying the two scores accordingly (Stecanella, 2019).

TF-IDF converts each token in the listing to a token weight based on how important that token is to the listing, normalized by the number of times the token appears in the whole corpus. This normalization helps to reduce the impact of common tokens in the collection. Finally, each listing is then converted to a vector of token weights. Equation (1) provides a process overview.

Text, letter

Description automatically generated

After cleaning our data to only include the most significant tokens and calculating the TF-IDF, we have a new dataframe that acts as an intermediate from our first dataset. This is because the prioritization of significant tokens likely caused some listings without significant tokens to completely disappear. Next, we create the index to separate our data into a training set and testing set, and convert these into dataframes so that they can be input into various modelling techniques, where each row in the dataframe is a product listing.

Chart, bar chart, funnel chart

Description automatically generated

Figure Five: Words most Relevant for Classifying Product Listings

These charts show the results of the bag of words approach and the TF-IDF model. On the left hand side are the terms that are most important to determining whether a product listing is classified as narcotics or other listing, with the words at the top of the list having higher TF-IDF values which signify a higher importance of the term in its role in classification. Some of the most significant words for the classification of a product listing as a narcotic include oxycodon (the most important), 5mg, and 220mg. For classification of a product listing as other listing, significant words include voo (the most important), fulz, and dvdrip.

Linear Discriminant Analysis (LDA):  
 We chose to implement an LDA regression classification method to find a linear combination of features that would characterize the two classes narcotics and non-narcotics. The resulting combination may be used as a linear classifier, or in our case, as a dimensionality reduction before performing the classification. LDA seeks out linear combinations of variables that best explain the data. It is defined as a dimensionality reduction technique since each successive transformation maximizes the between-class variance relative to the within-class variance. This procedure transforms the feature space to a linear space. After transforming the input data, new points can be classified by determining the closest centroid in the affine space under consideration of the class priors. We wanted to apply three different types of classification for the dataset, with the LDA approach being our shrinkage method that uses a linear decision surface in order to fit the data on a new axis. Since it is a supervised method it will also consider the outcomes which is a significant aspect for classification methods.

Naïve Bayes Classifier:  
 A Naive Bayes classifier is a supervised probabilistic machine learning model that’s used for classification problems, centered around Bayes Theorem. Bayes Theorem is a prediction tool which calculates the probability of a hypothesis given prior data. It is stated as P(h|d) = (P(d|h) \* P(h)) / P(d), where h represents hypothesis and d represents data. Essentially, the model relies on training data to create an accurate classification. It is called *naive Bayes* since each variable is assumed to be conditionally independent, allowing a simplified calculation of probabilities as P(d1|h) \* P(d2|H)\*...(Pdn|H) (Brownlee, 2020). When variable selection is implemented and properly introduced in the model, Naïve Bayes can perform as well as other statistical models such as logistic regression and SVM, even if the assumption of independence does not hold (Brownlee). Naïve Bayes, being a probabilistic classifier, will provide us a basis for comparison between other types of modelling techniques.

Feed-Forward Neural Network:   
 A feed-forward neural network has the goal to approximate some function f\* for a classiﬁer, y = f\*(x) maps an input x to a category y. The network deﬁnes a mapping of y = f(x; θ) and learns the value of the parameters (θ) that result in the best function approximation. Application of the feed-forward neural network can also overcome the issue of non-linearity. The neural network then trains a model by combining different distributions in one desirable distribution that considers the non-linearity of the problem (Gupta, 2018). The feed-forward neural network has high accuracy and a chance for greater predictability in the explanation of output by the model. It can serve as a contrast to the multinomial log linear model that also uses neural networks since the feed-forward method can serve as a non-parametric way of fitting the data to a model to the multinomial’s parametric technique.

Tuning hyperparameters:   
*I. Grid search approach*   
 The grid search utilizes an exhaustive approach for tuning hyperparameters defined in our non-shrinkage models of classification. Over a sequence of values, we coupled a 10 fold cross validation approach with three separate iterations on each model to optimize training accuracy. This means splitting the dataset into 10 groups 3 different times for each of our methods in order to evaluate the methods for bias and variance. Having 3 separate iterations across each fold allows for us to test for the highest accuracy levels according to our parameters and minimized overfitting. We choose to only optimize one hyperparameter in each of our models; this is mainly done to ease computational efforts. Unfortunately, iterative approaches across multiple hyperparameters are unrealistic for our group’s hardware; these approaches require significant time and memory that were unattainable for an approach outside of a simplified grid search across a single parameter. Future studies should actively seek to iterate over multiple parameters if they have the power required. We utilized the “caret” package to train and modify our hyperparameters iteratively.

*II. Naive Bayes Model*

We train our Naive Bayes model by modifying the Laplace correction value. Briefly, modifying the Laplace smoothing factor provides a sequential, iterative process for setting bounds on the conditional probability of a predictor (Yang, 2020). For our model, we sequenced values between the range 0 and 2 by values of 0.2 to track cross-validated accuracy of our model. Because most variation in the data can be captured by smaller values of our Laplace correction, we chose to utilize this range to optimize predictive accuracy. Although the “caret” package allows for changes to both the distribution and bandwidth of a naive bayes model, we opted to keep our iterative process simple for increased interpretability. Intuitively, our sequenced values for Laplace smoothing should ease cases in which term frequency result in zero probability for our assigned categories.

*III. Feed Forward Neural Network*

               For the neural network, we chose to sequence over the decay values while keeping the width of the network constant. In determining how to tune the model for cross validation, the network decay modulates the rate at which the neural network is able to respond to mistakes in the classification process. Utilizing the grid search approach, we cross validated the model at logarithmically sequenced decay values from10-4 to 10-1 derive the optimal tuning rate.

By finding the optimal network decay rate we are able to better train the model for classification. For the sake of simplicity, our neural network only has a single layer with no reiterative processes and approximately 9,000 nodes.  Further expansions of the neural network in either of these criteria would have made computations infeasible.

V. Results

Linear Discriminant Analysis:

Chart

Description automatically generated

Figure Six:Confusion Matrix for LDA-based Binary Classification

This is a confusion matrix of the linear discriminant analysis method of classification. It shows us that the model correctly predicts 657 listings as narcotics and incorrectly classifies 600 narcotic listings as non-narcotics. It also correctly classifies 492 non-narcotics as non-narcotics and incorrectly classifies 470 non-narcotics as narcotics. We observe that the accuracy of the baseline model is 0.5178 on the data with a 95% confidence interval of: (0.4968, 0.5388). The lower accuracy level observed for the LDA model likely reflects its inability to account for non-linearities and its linear decision space. This clearly contrasts with the feed-forward network which is a model that does an effective job factoring in non-linearities. Given the large feature size of the dataset, LDA may not perform as well compared to other algorithms. If future methods were to be employed, to correct this, principal component analysis could significantly reduce the dimensionality as a preprocessing step, therefore increasing accuracy.

Naive Bayes:

Chart, line chart

Description automatically generated

Figure Seven:Sequenced Cross-Validated Accuracy for Naive Bayes Classification

This graph demonstrates the training performance of the Naive Bayes classification method with its level of cross validation accuracy in relation to the Laplace correction value. According to the results, for Laplace correction values from 0 to 2 the cross validation accuracy of the model is the same at a value of about 0.727 on the testing data. The lack of change in accuracy from different Laplace correction values may be related to the structure of our data and we coded it into the TF-IDF model. We may be losing some variability in our categorical variables which are supposed to be the variables that are smoothed over by the Laplace corrections. With no variety in the variables to smooth over, a difference in Laplace correction values would potentially not result in any change in accuracy.

Chart, pie chart

Description automatically generated  
Figure Eight:Confusion Matrix for Naive Bayes Binary Classification

This is the confusion matrix for the Naive Bayes classification method. It shows us that the model correctly predicts 837 listings as narcotics and incorrectly classifies 420 narcotic listings as non-narcotics. It also correctly classifies 567 non-narcotics as non-narcotics and incorrectly classifies 395 non-narcotics as narcotics. This overall results in an accuracy of 0.6327 for the baseline model on the actual data with a 95% confidence interval of (0.6123, 0.6528). The model likely did not yield a higher accuracy since it works best under a strong assumption of independence between input variables, which we do not have. Still, the results of this method proved to be much more effective than LDA. This further suggests that the limitations of the LDA model is due to nonlinearity in the data.

Chart, line chart

Description automatically generatedFeed-Forward Neural Network:

Figure Nine:Sequenced Cross-Validated Accuracy for Feed-Forward Neural Network

This graph expresses the performance of the feed-forward neural network. As the network decay value increases, the cross-validated model accuracy decreases after the value 0.001. It reaches its highest accuracy of 0.887 for network decay logarithmically sequenced value of 0.001. Then the accuracy of the cross validation decreases for all values greater than the optimal network decay value.

The model correctly predicts 1143 listings as narcotics and incorrectly classifies 114 narcotic listings as non-narcotics. In addition, it also correctly classifies 836 non-narcotics as non-narcotics and incorrectly classifies 126 non-narcotics as narcotics. Overall, this  results in a cross validation accuracy level of about 0.887 for the feed-forward neural network method on the testing data. We observe that the accuracy of the baseline model on the data is 0.8918 on the data with a 95% confidence interval of: (0.8782, 0.9045). This model resulted in the highest accuracy level among the three classification methods tested, likely because of the single-layer neural network’s ability to account for non-linearities as well as the fact that it does not require independence between input variables.

Chart

Description automatically generated

Figure Ten:Confusion Matrix for Naive Bayes Binary Classification

VI. Conclusion

The classification methods tested in this paper attempted to solve a cumbersome issue undermining research and law enforcement efforts. From the three classification methods tested, we concluded that the feed-forward neural network is the most effective. The neural network sported an observed accuracy of 0.887 on training-set data and an accuracy of 0.8918 on the baseline model. Compared with Kinningham’s supervised learning methods, which maximally achieved an accuracy of 79% on a withheld test-set and an accuracy of 62% on a baseline model using soft margin SVM, neural networks may offer a more elegant solution for automatic classification.

There are several limitations when handling text-based classification problems that continue to affect the predictive accuracy of our data. The first limitation comes from the potential structure of the data. In future models, making sure that observations are relatively balanced may improve the predictive accuracy for any model. Unlike most real life cases, we cleaned the data to balance unique product listings between narcotics and all other listings. Future predictive models must consider weighting text matrices appropriately in the likely event of unbalanced categorical data. In addition to this, our text-based approach did not utilize any form of feature reduction before implementing supervised learning techniques. Text weighted matrices exponentially increase as product listings are fed into the data pipeline, and significant computational power is required to both parallelize and run calculations. Future models should implement feature reduction techniques, such as latent semantic analysis, in order to capture a significant amount of variation within the dataset to maximize computational power.

A final limitation from price listing data revolves around the veracity of the listing itself. Currently, dark markets are vast, decentralized networks that rely on anonymity. Accordingly, small vendors and a vast quantity of individual listings on dark markets are hard to verify. Although escrow services for buyers and sellers do provide an incentive for parties to enter into contracts, these services are not widely available. Distinguishing between real and fake product listings is a difficult task that can compromise the external validity of the model. If possible, future models should incorporate corrections accounting for data validity. While this paper mainly focuses on prediction, future models may care to focus on causal relationships between product classifications and established sellers.

Natural language processing is a powerful and robust tool for classification. As traffic on the dark web increases over time as a result of prioritizing anonymity, it is integral for researchers to have accurate, clean, and descriptive data. Utilizing machine learning algorithms to assist researchers has many benefits; minimizing human error and data scalability are only a few tangible benefits for text classification on the dark web. Although there is no uniform method for text classification, we believe that this workflow coupled with iterative cross validation provides a robust methodology for large-scale classification problems. Product listings are only the tip of the proverbial iceberg on the darknet; demystifying the wide array of services and goods the dark web offers provides insight on the motivations of those who use the dark web. As powerful text-processing algorithms continue to flourish, large-scale studies can accurately contextualize the underpinnings of a once-obfuscated environment with increasing ease. In an era of increased concern about one’s data footprint and digital security, this study hopes to establish an introductory, yet integral step in processing “big data”.

Bibliography

Boehmke, Bradley. “Creating Text Features with Bag-of-Words, n-Grams, Parts-of-Speach and More.” *Creating Text Features with Bag-of-Words, n-Grams, Parts-of-Speach and More · UC Business Analytics R Programming Guide*, 2018, uc-r.github.io/creating-text-features.

Branwen, Gwern. “Darknet Market Archives (2013-2015).” *Gwern.net*, 12 July 2015, www.gwern.net/DNM-archives.

Brownlee, Jason. “Naive Bayes for Machine Learning.” *Machine Learning Mastery*, 14 Aug. 2020, machinelearningmastery.com/naive-bayes-for-machine-learning/.

Graczyk, Michael, and Kevin Kinningham. “Automatic Product Categorization ForAnonymous Marketplaces.” *Semanticscholarship.org*, 2015, www.semanticscholar.org/paper/Automatic-Product-Categorization-for-Anonymous-Kinningham-Graczyk/1f7978d2e509c1a8ed049978f832ba2a3912a574.

Gupta, Tushar. “Deep Learning: Feedforward Neural Network.” *Medium*, Towards Data Science, 16 Dec. 2018, towardsdatascience.com/deep-learning-feedforward-neural-network-26a6705dbdc7.

Hvitfeldt, Emil. “Binary Text Classification with Tidytext and Caret.” *Emil Hvitfeldt*, 1 May 2020, www.hvitfeldt.me/blog/binary-text-classification-with-tidytext-and-caret/?fbclid=IwAR3YRAB4hcrIsPs5RxNLfPLC4yWsMdjSNBFPifitDHuk84Ppa5NITrUxQLs.

Stecanella, Bruno. “What Is TF-IDF?” *MonkeyLearn Blog*, 10 May 2019, monkeylearn.com/blog/what-is-tf-idf/.

Stecanella, Rodrigo. “The Beginner's Guide to Text Vectorization.” *MonkeyLearn Blog*, 21 Sept. 2017, monkeylearn.com/blog/beginners-guide-text-vectorization/.

S., Yang. “An Introduction to Naïve Bayes Classifier.” *Medium*, Towards Data Science, 8 May 2020, towardsdatascience.com/introduction-to-na%C3%AFve-bayes-classifier-fa59e3e24aaf.