Paper Title\* (use style: paper title)

Eschholz, Axel  
*Jodrey School of Computer Science*  
*Acadia University*Wolfville, Canada  
161044e@acadiau.ca

*Abstract*—This electronic document is a “live” template and already defines the components of your paper [title, text, heads, etc.] in its style sheet. *\*CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract*. (*Abstract*)

Keywords—component, formatting, style, styling, insert (key words)

# Introduction

In an economy increasingly driven by e-commerce, optionality for consumers has never been greater [SOURCE]. Furthermore, the structure of online reviews allows consumers unprecedented access to information about prospective products and services. In order to compete in the digital age, companies must be able to analyse these customer reviews to evaluate and improve their digital image and reputation. In the context of “stars” or a rating system, this is a fairly trivial matter. However, when no such linear system exists, it becomes a much more challenging and ambiguous task. To properly categorize reviews at scale, a potential solution is to employ sentiment analysis to ascertain the relative quantity of positive to negative reviews. Due to the subjective and specific nature of sentiment analysis, this is a potential application for a machine learning model. Therefore, this project will aim to create a model to classify online customer reviews. To restrict the investigation scope, this project with only explore the potential for decision-tree models.

This model will be constructed from one such dataset of unlabelled customer reviews sourced from TrustPilot, an online review platform. These reviews pertain to a company called ASOS. Given that the model will be used to drive aggregate information about a large corpus of data, success criteria for this project will be creating a model that can classify consumer reviews as ‘positive’, ‘negative’, or ‘neutral’ with an accuracy of at least 80%.

# Background

## Sentiment Analysis

Sentiment analysis is a field focused on identifying and analyzing subjective information contained in text data. It aims to classify the sentiment behind a piece of text, categorizing it as positive, negative, or neutral, and sometimes even more granular emotions like happiness or anger. Sentiment analysis therefore often leverages Natural Language Processing (NLP) to extract subjective and contextual meaning from text [1].

Given the subjective and inexact nature of sentiment analysis, constructing accurate models can be extremely difficult, especially when contending with more nuanced sentiments. Yet even with a more simplistic positive/negative scale, creating a generalizable model can prove elusive. This challenge is often exacerbated by shifting context. When sentiment can refer to multiple contexts words might carry different sentiments in different settings. This variability makes it difficult to create universally effective models [2].

## Decision Trees

Decision trees are a type of machine learning algorithm used for both classification and regression. They are used to model decisions and their possible outcomes with a tree-like structure, with decision nodes branching downwards from a root node to terminate in decision values, aptly referred to as ‘leaves’. Decision trees are trained by attempting to find the attributes that deliver the greatest information gain at each node in pursuit of producing the most accurate tree with the least number of nodes. By ‘pruning’ the tree to reduce the depth, they can be generalized to broader datasets.

A significant motivation for using decision trees is their transparency. Many machine learning approaches, like Artificial Neural Networks (ANNs), are extremely opaque in the resulting decision process, thereby making them difficult to troubleshoot and refine. Decision trees, on the other hand, can often be visualized and evaluated after training, a significant advantage for refining models [3].

## Lexicon-Based Labelling

This project will employ a lexicon to label the training data. A lexicon is a corpus of words and phrases associated with a sentiment value on a certain scale. One such lexicon is the AFINN lexicon. Developed at the Technical University of Denmark in 2011, the AFINN lexicon was specifically developed for sentiment analysis of “microblogs”, or short chunks of text casually posted online [4]. AFINN was developed using twitter posts from 2009, then refined against leading lexicons at the time such as ANEW.

## Term Frequency & Inverse Document Frequency

Term Frequency - Inverse Document Frequency (TF-IDF) is a method used to determine the relative importance of a given token within an instance. This value is derived from the number of times a token appears in a given instance in relation to the relative frequency of that token across the dataset [5]. When applied, it turns a token list into a set of *n* vectors, each representing a specific token, where *n* is the specified dimensionality of the vectorizer. These vectors then become attributes for the transformed data to be fed into the model.

# Method & Approach

## Project Plan

This project can be broken down into three distinct problems. First, the training data must be properly labelled to facilitate supervised learning. Then, the data must be cleaned and transformed into a format that can be modeled. Finally, the appropriate model must be built and refined using these preprocessed data.

To label the training data, a combination of a manual and lexical labelling will be employed. Approximately 10% of the given data will be initially labelled manually. Then the AFINN lexicon will be applied across all instances to the review title and review contents separately. The relative distribution of the scores for labelled instances will be analysed, moderating for the given label, to determine logical boundaries for the classification. These boundaries will then be applied across all instances to label the training data.

To transform the data into a workable format, TF-IDF vectorization will be applied to the preprocessed tokenized data. This will yield a set of features that represent important tokens across the dataset.

Finally, to build and refine the model, attribute selection will first be applied to the TF-IDF results to significantly reduce dimensionality. Once this has been performed, the selected decision tree model will be applied and then fine-tuned to produce the most promising result.

## Preprocessing

The initial state of the data was 2000 unlabeled reviews in plain text. To clean the data, a tokenizer was first applied to break the text into discrete words. Frequently occurring words with little to no significance in the context of sentiment analysis are known as stopwords. Removing these stopwords from the token list can help reduce noise and improve the accuracy of the model [6] In this instance stopwords are removed using a set of common stopwords provided by the nltk library in python.

The next step in preprocessing is to apply TF-IDF vectorization to the remaining to derive numeric attributes that will then be used in classification. In this instance the vectorizer generates 1000 attributes per instance of the most important tokens in the dataset.

## Modeling

Given the restriction of the project scope to decision tree models, the optionality for potential models is naturally limited. Three common approaches to decision trees include the C4.5, random tree, and random forest algorithms.

The C4.5 Algorithm represents a more classic approach to the creation of decision trees, using relative information entropy and gain for attribute selection to produce a reliably accurate decision tree [7]. The C4.5 algorithm benefits from fast computation and relative transparency in the classification process, but handling datasets with high dimensionality often necessitates further attribute selection.

Random tree algorithms present a vastly different approach to the methodical C4.5 algorithm by splitting nodes on a random subset of features. While fast to train, their high variance makes them too unreliable for individual use. Random forest algorithms solve this issue by creating batches of random trees and aggregating their results. In doing so, random forests address the variance issues of random trees and produce highly accurate models. The structure of random forests enables effective handling of high dimensionality without the need for prior attribute selection. However, these benefits come at the price of transparency, as the classification process of random forests is extremely opaque. Random forests also use a significant amount of computing power, and therefore take longer to train, compared to other options. [SOURCE]

Despite the potential drawbacks of the random forest approach, the extremely high dimensionality of the prepared data in this case makes it a potentially more accurate and generalizable option. This project will investigate the relative accuracy of each model in this application.

## Evaluation

Cross-validation will be used in the training of the initial model. The model will then be evaluated on a holdout test set of 100 examples labelled in the same way as the training examples. The accuracy of the holdout test set evaluation will indicate the relative generalizability of the model and reflect any potential overfitting happening during the training process.

# Results

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

## Authors and Affiliations

**The template is designed for, but not limited to, six authors.** A minimum of one author is required for all conference articles. Author names should be listed starting from left to right and then moving down to the next line. This

Examples include Acknowledgments and References and, for these, the correct style to use is “Heading 5”. Use “figure caption” for your Figure captions, and “table head” for your table title. Run-in heads, such as “Abstract”, will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head

#### the abbreviation “Fig. 1”, even at the beginning of a sentence.

1. Model Application Results

| ***Run*** | Model | Hyperparameters | Accuracy | Holdout |
| --- | --- | --- | --- | --- |
| 0 | J48 | -C 0.25 -M 2 | 67.54 | 69 |
| 1 | RF |  | 73.49 | 67 |

1. Sample of a Table footnote. (*Table footnote*)

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity “Magnetization”, or “Magnetization, M”, not just “M”. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write “Magnetization (A/m)” or “Magnetization {A[m(1)]}”, not just “A/m”. Do not label axes with a ratio of quantities and units. For example, write “Temperature (K)”, not “Temperature/K”.

##### Discussion

The limitations of this investigation are significant, ranging from oversimplified tokenization to

##### Conclusion

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

##### References

1. Panchal, D., Mehta, M., Mishra, A., Ghole, S., & Dandge, M. (2022). Sentiment Analysis Using Natural Language Processing. International Journal for Research in Applied Science and Engineering Technology.
2. Shayaa, S., Jaafar, N., Bahri, S., Sulaiman, A., Wai, P., Chung, Y., Piprani, A., & Al-garadi, M. (2018). Sentiment Analysis of Big Data: Methods, Applications, and Open Challenges. IEEE Access, 6, 37807-37827.
3. Kotsiantis, S. B. (2013). Decision trees: a recent overview. *Artificial Intelligence Review*, *39*, 261-283.
4. AFINN Lexicon
5. Abubakar, Haisal Dauda, Mahmood Umar, and Muhammad Abdullahi Bakale. "Sentiment classification: Review of text vectorization methods: Bag of words, Tf-Idf, Word2vec and Doc2vec." *SLU Journal of Science and Technology* 4.1 & 2 (2022): 27-33.
6. Ghag, K. V., & Shah, K. (2015, September). Comparative analysis of effect of stopwords removal on sentiment classification. In *2015 international conference on computer, communication and control (IC4)* (pp. 1-6). IEEE
7. Meena P I M and Perumal V 2016 Performance of C4.5 and Naïve Bayes Algorithm to Predict Stomach Cancer - An analysis. 5 482-87
8. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
9. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
10. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.