Classifying Clickbait Titles – A Supervised Learning Approach

Mark Biegel — May 27th, 2022

Abstract

This report details an approach to combating clickbait using multiple supervised machine learning techniques. With the introduction of online media, clickbait has become an ever-growing issue. Publishers are able to earn a profit based on how many users visit their site; this has resulted in the widespread use of misleading titles. In the experimentation, data sets are implemented on four classifiers to compare each for detecting clickbait: Multinomial Naive Bayes, Stochastic Gradient Descent Classifier, Perceptron, and Support Vector Machine. Accuracy, F1, and ROC area under curve scores are calculated on each classifier to showcase how well each did detecting clickbait. All four classifiers are used in tandem to determine if a title is clickbait by averaging each classification produced and making an overall clickbait status. Overall, the experimentation proves that supervised machine learning is an effective way to determine article and video clickbait status.

1. Introduction

The internet is the 21st Century's almanac, with information easily accessible at everyone's fingertips. Books, articles, and videos on the internet provide quality and beneficial sources of information for public consumption; however, these technological advancements have drawbacks. While sites provide relevant articles and videos, they can use misleading titles that exaggerate the true content of the story in order to increase the likelihood of users clicking on their site. This paper details the utilization of machine learning to decipher and detect clickbait titles of articles and videos present on the internet.

1.1. Defining Clickbait

Misleading titles for news-worthy stories is not a new idea. From the early days of newspapers, hyperbolized titles have been used by printing companies to embellish stories in order to sell more copies; the internet has only made this more frequent and easily accessible. In the modern era, clickbait is the term for deceptive titles that are misleading to the content of an article or video. While there are endless possibilities, clickbait titles follow a certain trend with its composition. Often, titles regarded as clickbait have exaggerated words or phrases that hyperbolize the true content of the story. Sometimes, titles involve second person pronouns to directly associate the emotion in the title to the user. Because of clickbait, the phrase "Don't believe everything you read on the internet" became popular since it is a way to remember that information is easily manipulated and falsified on the internet. Even with the existence of this phrase, people still fall into the clickbait trap.

1.2. Clickbait Today

Many online publishers of clickbait have been ridiculed for having misleading titles that exaggerate the story in order to get clicks. This is done because publishers' sites earn money to have advertisements overlaid on videos or articles. The advertisers pay the site-holders by how many users visit, or click on, the site, so the more clicks a site receives, the more profit a publisher receives. Thus, site-holders want their article and video titles to be as catchy as possible in order to get users to click on them. This has led to a plethora of misleading and inaccurate titles for many articles and videos found throughout the internet, with some users not even aware of how widespread clickbait is.

There are organizations, such as Stop Clickbait that aim to expose stories with clickbait headlines. **Figure 1** is an example of Stop Clickbait exposing a story with a misleading title. The title of this story builds up to the idea that there is something truly earth-shattering with the flavor of green gummy bears. As Stop Clickbait points out in the image, however, the story is simply that green gummy bears are strawberry-flavored; hence, this detail could have been defined in a simple statement without an overemphasized title.

With misinformation and exaggeration plaguing online media, society could greatly benefit with a way to determine clickbait titles. Machine learning offers a great way to predict future events through use of pattern recognition in data analysis. This report entails the use of supervised machine learning techniques to detect clickbait titles.

Figure 1. **Stop Clickbait exposing a clickbait article**

2. Purpose

2.1. Motivation

Clickbait is everywhere on the internet today, so it is difficult to determine if article and video titles are overemphasizing the magnitude of the information discussed in the story before clicking on it. It could be exceptionally useful to have an algorithm that can check if titles are considered clickbait, so users do not have to waste time reading or watching distorted stories.

Furthermore, even if users do not mind visiting clickbait web pages, clicking on sites involuntarily makes publishers money since the page contains ads that earn revenue from each user who views the site. If wasted time is not an important issue to users, not clicking on misleading sites to lessen the amount of money publishers make is another incentive to have a clickbait algorithm.

Nonetheless, clickbait exists in political articles and videos. Contrary to misinformation that also exists in politics, clickbait can be more easily deciphered compared to misinformation. In the political realm, clickbait has exaggerated and emotionalized titles that can smear a specific group or person. These titles tend to be opinionated and contain slanderous terms that are clickbait. Having a detection algorithm can deter users from viewing that site in order to disincentivize the use of clickbait.

Finally, having some clickbait detection algorithm can benefit users by enabling them to be more aware of the the titles to stories and videos they click on. Even if the clickbait classifier is not extraordinarily accurate, the classification can give users a baseline observation as to whether the article or video they are about to view is considered clickbait or not. Over time, users can learn to recognize clickbait on their own, allowing them to make more informed decisions about embellished titles, becoming a more robust internet user.

2.2. Problem Statement

Currently, there is no robust method to determine if an internet title is clickbait or not. Users have to make decision to the best of their knowledge about a title being clickbait; some users may not even be aware of how common clickbait is on the internet.

Given that there is no easy way to classify titles as clickbait, a comprehensive solution should be implemented to detect clickbait articles and videos before users view them. One solution is to use machine learning. Supervised machine learning on clickbait title dataset offers a robust algorithm that classifies internet titles as clickbait, helping users avoid misinforming and time-wasting pages. This can help purge the internet of bothersome media that provide little value to user consumption, and it enables users to be more robust when navigating the internet.

3. Methodology

There are numerous ways to create a clickbait classifier. The experimentation in this report consists of supervised learning techniques that are used in coalition to predict classification statuses of article and video titles. Natural Language Processing (NLP) is the core of this experiment, as the input data is text-based; thus, the learning models chosen should be effective for classifying text-based tasks.

3.1. Supervised Learning

Supervised machine learning is a convenient method for prediction classifiers. Supervised learning takes pre-existing, labeled data to train a model in which the model can use to classify other data and make predictions based on its training (1). Supervised learning has helped in a multitude of areas for prediction and classification. Stock market value, weather forecast, housing prices, or any field that contains a plethora of collected data that has been documented and labeled can be used in supervised learning models.

The methodology uses four models. Each model has a different functionality for classification, so using all four models in tandem creates a broader and more accurate classification status compared to solely using one. The four classifiers used are:

- Multinomial Naive Bayes
- Stochastic Gradient Descent Classifier
- Perceptron
- Support Vector Machine

These four classifiers were chosen because of prior familiarity with implementation as well as popularity of the models for NLP. During the evaluation of each model, accuracy, F1, and Receiver Operating Characteristics area-under-curve (ROC AUC) scores are calculated. Accuracy is determined by taking the correctly classified titles and dividing by the total number of titles in the test set. F1-score uses precision and recall to find a mean between the two metrics. A Receiver Operator Characteristic (ROC) curve is plotted to show how well the model can identify clickbait by comparing the relationship between true positive and false positive rates. Directly related, the area under the ROC curve demonstrates how well the classifier can depict between two classes; maximizing ROC AUC translates to a better performing model. Together, these metrics showcase how well a model is able predict the correct class.

3.1.1. Multinomial Naive Bayes

Naive Bayes classifiers have a variety of different models that pertain to specific use-cases. Multinomial Naive Bayes is a modeling algorithm used for natural language processing, as it works well with classifying discrete features that are independent of one another.

This model is chosen for clickbait detection because of its popularity with language processing and because this experiments utilizes text interpretation of titles the model trains on.

3.1.2. Stochastic Gradient Descent Classifier

In contrast to Multinomial Naive Bayes, Stochastic Gradient Descent (SGD) is merely an optimization technique to train a model, but it is not associated with a specific model. Thus, the Stochastic Gradient Descent Classifier is implemented, as its foundation is the standard stochastic gradient algorithm that uses a variety of loss functions to linearly separate data. As with machine learning in general, each loss function produces a different accuracy depending on the data it receives, so each implementation of Stochastic Gradient Descent must be fine-tuned to determine the most appropriate loss function for the inputted data.

This classifier is chosen for clickbait detection because of its popularity for supervised natural language processing classification. Stochastic Gradient Descent is very versatile, and it is proven to be an effective model for large projects involving NLP, as it is able to scale on extensive datasets.

3.1.3. Perceptron

Perceptron is a supervised learning model that is suitable for large-scaled datasets. It does not require a learning rate, does not involve penalization, and it updates the model based only on mistakes. These are useful features because it speeds up runtime and increases overall efficiency. While not the same, Perceptron is comparable to the SGD Classifier because the underlying implementation is similar; however, Perceptron can produce sparser models than the SGD Classifier in less compute time. Thus, its overall performance is better than the SGD classifier.

This model is chosen for clickbait detection to directly compare the results with the Stochastic Gradient Descent Classifier in order to quantify if and how much better Perceptron is compared to SGD for clickbait classification.

3.1.4. Support Vector Machine

Support Vector Machines (SVM) offer the ability to take data into high-dimensional feature spaces with sparse feature data; this enables SVMs to effectively linearly separate data to make a classification. Because of its high-dimensionality characteristics, SVMs are useful for NLP tasks. Furthermore, it allows flexibility with kernel customization for improved accuracy and overall effectiveness. A downside to SVMs is that a large number of samples trained on model can result in expensive runtimes.

This model is chosen for clickbait detection because SVMs are robust models that have worked well in other natural language processing applications. Furthermore, an SVM can be directly compared to the other classifiers in terms of runtime since SVMs can be slow on large datasets.

3.2. Dataset

For supervised machine learning, the training dataset is the foundation for an effective model. An exhaustive dataset that reflects real-world representation as well as specific edge cases conditions a model to be robust and effective. A perfect sample dataset does not exist, as there can be implicit bias amongst the values; however, with a large and diverse dataset, a general solution can be formed with as little bias as possible that reduces the risk of over-fitting to the sample data.

In order to find robust datasets, the Kaggle® dataset search engine is used. From Kaggle®, six potential dataset involving clickbait emerge; however, the most robust datasets involve specific classification of titles as either clickbait or not clickbait. The two datasets that are used to train the models in the experiment are Clickbait Dataset and YouTube Clickbait Classification. Clickbait Dataset is a compilation of general clickbait titles found across the internet and is used to train the classifiers. This dataset contains over 32,000 individual titles: about 16,000 titles are classified as clickbait with the remaining 16,000 titles classified as not clickbait; thus, a robust model is generated due to the large amount samples for training. On the other hand. YouTube Clickbait Classification are specific clickbait titles gathered

on the YouTube platform and are used for testing the classifiers. This dataset is solely predicted on by the models and not used for training. Accuracy, F1, and ROC AUC scores are produced from the models based on this dataset, demonstrating how well the models classified external titles.

Overall, the compiled dataset is more than enough to get accurate results on the effectiveness of classifying clickbait titles with supervised learning.

3.3. Intuition

With the previously declared classifiers and dataset defined, supervised learning works because of its proven ability to train models on labeled data in order to classify other sources of data. The classifiers have been specifically chosen for this experiment since all four are very good with natural language processing tasks. Furthermore, the compiled dataset contains thousands of labeled titles that are generalized to portray a variety of article and video titles on the internet. While the classifiers might not be the best choice for the task and while the datasets might not generalize all types of clickbait titles, these tactics test the usability of supervised learning on clickbait classification.

4. Experimentation

The goal of the experiment is to determine if supervised learning is effective at clickbait classification. In the process, because there are four classifiers in use, the experiment should compare the results of each classifier against one another to see which is the most accurate. This is done by calculating a model's accuracy and F1-score. A secondary goal is to be able to allow for real-time user input, so users can enter titles to know if it is classified as clickbait.

With the methodology defined, a Python program is written to test the the proposed methods and to see how effective supervised learning is at clickbait detection. Pandas, NumPy, and Scikit-learn modules are used in the process for loading data, formatting data, and extracting features. The data is stored in a CSV file, so it can be loaded into the application via a Pandas Dataframe. The test bed is created in a Jupyter Notebook for intuitive flow and easy navigation of the process. The Python program can be located at the following repository:

Clickbait Detection Application - GitHub Repository

4.1. Process

With a test bed defined, there are key components needed for Scikit-learn models to be able to make a classification based on the data.

4.1.1. Data Import

The CSV files for training and testing data contains two columns: a headline column of the titles and a classification column with the pre-determined clickbait classification. From the CSV files, the data is read in by each column and stored in a Pandas dataframe for easy manipulation throughout the program; pandas dataframes are useful for feature extraction. With training and testing data in their respective dataframes, natural language processing can be applied to the data for use in the models.

4.1.2. Feature Extraction

Part of the procedure for Natural Language Processing requires specific data to be chosen and converted into a numerical representation for the classifiers; this is known as feature extraction.

Training the classifiers requires quantitative sample data. Consequently, the data stored in the CSV file contains a list of titles as strings. Scikit-learn contains a feature extraction module called CountVectorizer() that is able to convert strings into a series of unique, numeric tokens that the models can train on. Vectorization is performed on the list of strings from the CSV file and is subsequently converted into a Document Term Matrix (DTM). The classifiers can be fitted to the DTM, as each can learn from the quantitative data representation of the titles. Nonetheless, the same vectorization process is applied to the test data, as the classifiers can only compare between quantitative data during prediction. With the data quantified, the models are created and trained for prediction.

4.1.3. Model Training and Prediction

All four models are created with specific, fine-tuned parameters to achieve the highest metrics:

- Multinomial Naive Bayes is created with the smoothing parameter $alpha$ set to 0.00001. This value is chosen over the default value of 1.0.
- Stochastic Gradient Descent Classifier is created using the $huber$ loss function with $max\ iterations = 5000$
- Perceptron is created using default parameters
- Support Vector Machine is created with a Support Vector Classification (SVC) model, using a sigmoid kernel function with $gamma = 3$.

Each model is fitted on the same training set. After being fitted, the models run predictions on the test data. Here, the models are given only titles and return a prediction on if the title is clickbait. This prediction classification of each is compared with the pre-determined classification, and the

resulting scores are quantified with metric modules from Scikit-Learn. Accuracy is calculated by totaling how many of the predicted classifications are correct and dividing it by the total predictions made:

$$Accuracy = \frac{correct\ prediction}{total\ predicted}$$

The F1-score is calculated using precision and recall in order to find a weighted average between the two:

$$F1 = \frac{2 * (precision * recall)}{precision + recall}$$

An ROC Curve is plotted to calculated the AUC, helping evaluate the classification ability for each model.

With accuracy, F1-score, and ROC curves, all metrics are gathered and can be analyzed against one another to draw conclusions about clickbait classification using supervised learning.

4.1.4. Secondary Goal - Real-time User Input

In addition to determining supervised leaning's capability, allowing titles found online to be entered into the test bed for classification enables users to avoid clickbait and learn to detect it for future reference.

Because the test data is what the user enters, the training dataset and test dataset are compiled together to make one large training dataset, increasing the samples the models learn from and increasing classification accuracy. This allows for more generalized models since video clickbait titles differ from article titles. Once the models are trained, the user is prompted to enter the title they want to classify, and then a prediction from each classifier is computed.

With four predictions, a final classification status is calculated based on a clickbait threshold. In this experiment, the threshold for a title being clickbait is if three of the four (or 0.75) classifiers predicted true; anything less than three of four clickbait predictions is considered not clickbait. This threshold is set at 0.75 for simplicity, as the ratio implies a majority of the models predicted the title as clickbait; there is room to fine-tune this threshold in the future.

4.2. Results and Discussion

With the test bed created, the respective metrics are generated based on the testing data.

4.2.1. Scores

The accuracy and F1-score for numerous runs are averaged and displayed in the **Table 1**.

The results demonstrate that Multinomial Naive Bayes performed the best out of the four classifiers. Stochastic Gradient Descent slightly edged out the more efficient Perceptron model, and the Support Vector Machine model did the worst out of the four; however, overall, all models performed adequately for clickbait classification

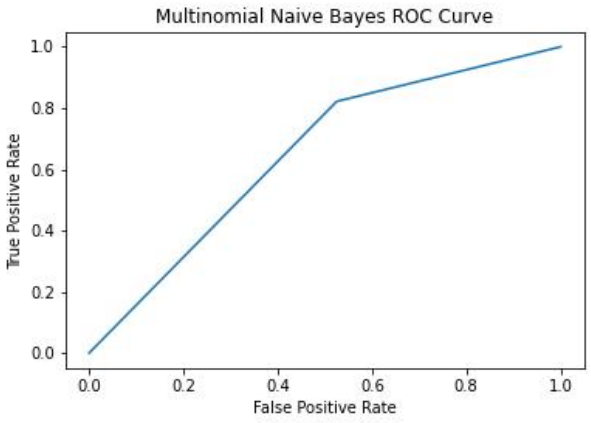
Classifier	Accuracy	F1-Score	ROC AUC
Multinomial Naive Bayes	0.6500	0.6383	0.5902
Stochastic Gradient Descent	0.6350	0.6301	0.5902
Perceptron	0.6100	0.6086	0.5902
Support Vector Machine	0.5900	0.5900	0.5902

Table 1. **Accuracy, F1, and ROC AUC scores per classifier**

4.2.2. Multinomial Naive Bayes

With an accuracy of 0.6500 and an F1-Score of 0.6383, Multinomial Naive Bayes performed the best out of all models. ROC AUC results in a value of 0.5902, indicating that the classifier is sufficient at classifying clickbait titles. The ROC Curve for Multinomial Naive Bayes is shown in Figure 2.

Figure 2. **Multinomial Naive Bayes ROC Curve**



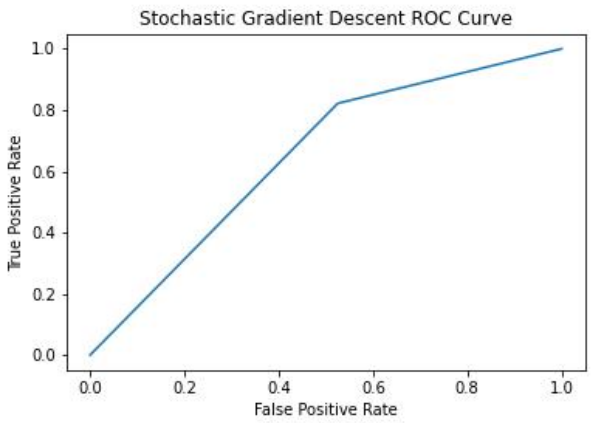
4.2.3. Stochastic Gradient Descent

With an accuracy of 0.6350 and an F1-Score of 0.6301, Stochastic Gradient Descent performed the second best out of all the models. ROC AUC results in a value of 0.5902, indicating that the classifier is sufficient at determining clickbait titles.The ROC Curve for Stochastic Gradient Descent is shown in Figure 3.

4.2.4. Perceptron

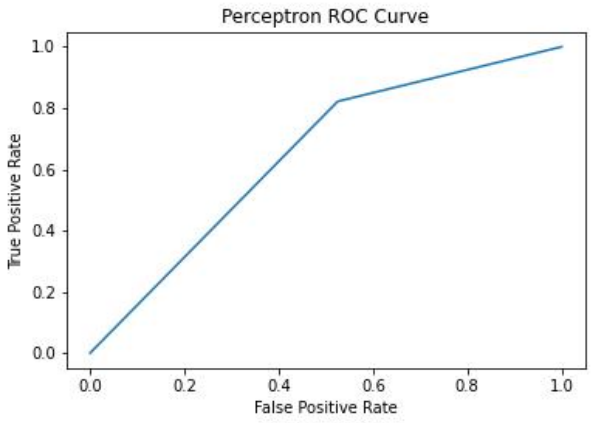
With an accuracy of 0.6100 and an F1-Score of 0.6086, Perceptron performed similarly SGD. ROC AUC results in a value of 0.5902, indicating that the classifier is sufficient at classifying clickbait titles. The ROC Curve for Perceptron

Figure 3. **Stochastic Gradient Descent ROC Curve**



is shown in Figure 4.

Figure 4. **Perceptron ROC Curve**



4.2.5. Support Vector Machine

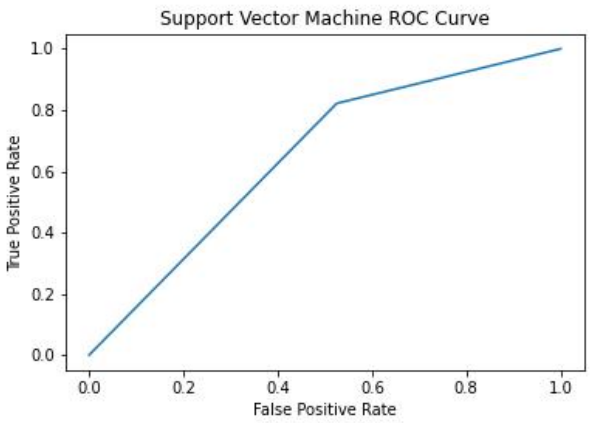
With an accuracy of 0.5900 and an F1-Score of 0.5900, Support Vector machine performed the worst out of all the models; however, it achieved the same ROC AUC value of 0.5902, indicating that SVM is just as sufficient at classifying clickbait titles. The ROC Curve for Support Vector Machine is shown in Figure 5.

4.2.6. Analysis

All four classifiers achieved very similar accuracies with the same ROC curve score in the experiment. This demonstrates that the classifiers equally performed in clickbait detection given the training and testing data.

The success of Multinomial Naive Bayes accuracy score can be directly attributed to the fact that it is designed for NLP classification; thus, it creates a better generalized model to classify clickbait titles. Stochastic Gradient Descent's accuracy came within a reasonable marginal to Multinomial Naive Bayes, showcasing how versatile of an algorithm it is for NLP classification. While Perceptron performed reasonably well out of the classifiers, a possible reason it fell behind SGD because it uses the hinge loss function only, while SGD has a variety of loss functions to choose from that can be fine-tuned it to fit the data more generally. Lastly, Support Vector Machine's lesser performance compared to the other classifiers could be attributed to inadequate parameters during the setup for the classifier. Overall, all classifiers produced reasonable scores, and because their ROC AUC scores were the same, it indicates that all these classifiers performed within error of each other. All models are effective at detecting clickbait in article and video titles.

Figure 5. **Support Vector Machine ROC Curve**



4.2.7. Model Improvements

While the results from the experiment are conclusive, the models can always be improved upon. Some tactics include better feature extraction from the data. A different feature extraction process could be used on the data to see if there is a more effective approach for quantifying strings. Furthermore, a larger dataset can improve prediction models. While 32,000 data points is more than adequate to train models, having different types of headlines from a variety of sources will improve on the generalization of each model. Nonetheless, different classifiers as well as different parameters for classifiers may improve the overall accuracy of clickbait detection. Some classifiers are more appropriate for NLP tasks, and specific parameters result in a more accurate and reliable model. Further research and experimentation may lead to a better overall clickbait classifier; however, the current experiment shows that supervised learning is able to

detect clickbait titles a majority of the time.

4.2.8. Secondary Goal - Real-world Performance

While the real-time user input secondary goal cannot be scientifically analyzed, its results should mimic the conclusions found in the main experiment. News board sites such as AOL.com and MSN.com were visited, as these pages contain a mix of stories from well-known companies as well as small publications. From here, a variety of titles were tested, ranging from factual titles that are more likely not clickbait to emotionally-exaggerated titles that are more likely clickbait. The results are comparable to the accuracies found above: more than half the time, the total classification status is able to determine an emotionally-exaggerated title as clickbait. In addition, the classification status never wrongly considered a factual title as clickbait; it only wrongly classified clickbait titles as not clickbait. This demonstrates that the classifiers are able to determine a factual title as not clickbait. Conclusively, the real-time input is an effective way for users to detect the majority of clickbait titles.

5. Conclusion

This report proposes an approach to detect clickbait among internet article and video titles using supervised machine learning methods. Multinomial Naive Bayes, Stochastic Gradient Descent, Perceptron, and Support Vector Machine algorithms are the models used to showcase the ability of supervised learning. Each model has fine-tuned parameters that give them the best performance for classifying clickbait titles correctly. Two large datasets of pre-determined clickbait titles are used to train the models during experimentation, capturing a wide variety of clickbait formats that facilitate generalized prediction models. Feature extraction is performed on the datasets in order to quantify the titles, so the models can train on the sample data. Accuracy, F1, and ROC AUC scores are calculated to rank the classifiers based on quantitative values. The Multinomial Bayes classifier resulted with the highest accuracy at 0.6500 and F1-score at 0.6383 compared to the other classifiers. While it had the same ROC AUC score as the rest, the slight advantage in the other metrics prove its design is superior for discrete feature classification in text applications. Furthermore, the secondary goal of a real-time user input program proves to be advantageous, as it performed with similar accuracy in comparison with the main experiment; this gives users an application to detect clickbait in an article or video title they come across.

In conclusion, supervised machine learning techniques are successful with clickbait classification, and with proper implementation, a classification application can be accessible for public consumption in order to combat the escalating issue of clickbait plaguing modern-day media.

References

[1] Suhaib R. Khater, Oraib H. Al-sahlee, Daoud M. Daoud, and M. Samir Abou El-Seoud. Clickbait detection. In *Proceedings of the 7th International Conference on Software and Information Engineering*, ICSIE '18, page 111–115, New York, NY, USA, 2018. Association for Computing Machinery.