Predicting the Appearance of Cited Article on Social Media Site-YouTube

Priyanjani Chandra[†]
Department of Computer Science and Engineering
Northern Illinois University
Dekalb, Illinois
Z1864520@students.niu.edu

Sneha Ravi Chandran[†]
Department of Computer Science and Engineering
Northern Illinois University
Dekalb, Illinois
Z1856678@students.niu.edu

ABSTRACT

The goal of the project was to predict the appearance of the scholarly article on social media sites, YouTube in specific. The Altmetrics dataset was taken into consideration. From the dataset, the necessary features and labels were chosen which helps in making a prediction as to whether the article is posted on YouTube or not. The feature selection involved in extracting the JSON file and analyzing the required Key-Value pair that would be used later for the further processing. The class labels/ target was the YouTube column taking two values, 0 corresponds to articles cited not present int YouTube and 1 corresponds to the article getting posted on YouTube. The dataset was divided into training and test set which lead to an imbalanced class. The class imbalance problem was resolved by taking into consideration the SMOTE, XGBoost, oversampling and undersampling. The models that chosen to perform the classification was the LR, SVM, Decision Tree, Random Forest, amongst which the one that yielded reliable accuracy was the Logistic Regression model with SMOTE and oversampling.

ABBREVIATIONS

•LR- Logistic Regression •SVM- Support Vector Machine •SMOTE- Synthetic Minority Oversampling Technique •XGBoost- eXtreme Gradient Boosting •ROC- Receiver Operating Characteristic Curve •TPR- True Positive Rate • FPR- False Positive Rate • TP- True Positive • TN- True Negative • FP- False Positive • FN- False Negative

KEYWORDS

YouTube citations, Altmetrics, Overfitting, Normalization, Logistic Regression, Random Forest, Decision tree, SVM, ROC, Precision Recall curve, Class imbalance, SMOTE, XGBoost, oversampling, undersampling, StandardScaler, Scikit Learn library, Matplotlib, pandas, dataframe, seaborn, overfitting.

1 Introduction

To potentially identify whether any particular research paper is posted on the social media platform based on related feature selection from the chosen dataset. Firstly, we should know how the articles are perceived on any given social media platform. In order to gain a fair knowledge about the comprehensive details such as how the cited papers are being perceived amongst a group , how the viewers enjoy the content or how they are being used as a reference for other related research proposal, we are considering YouTube as our choice of preference from the altmetric dataset and analyzing the features to classify them based on the appearance it had received on that platform. Thereby, having a tangible effect on benefiting the author to know how many users are engaged and exposed to any scholarly article on YouTube.

While blogs, Facebook, Twitter, Mendeley, etc., had a very high scope getting noticed among the research scholars, the contemplation for the videos getting posted grabbed our attention to make the prediction. The highly correlated features were not considered thereby not biasing towards any specific features.

2 Literature Review

The purpose behind altmetrics is to unite all the article-level metrics [1] that refers to the same publication. Altmetrics allowed us to know the academic use of various online environments. The use of altmetrics [2] properly can help in broader reach of new scientific innovations to the public. Altmetrics has been the source for computing alternative metrics on Twitter, blogs, Wikipedia, etc.

In the era of digitizing the search for the cited articles [3], we are looking at how any relevant research articles are posted on YouTube platform from that of the other social web services. This is chosen because, in real-time, the citation of any article takes a relative amount of time to be cited in any other research published or blogs as such but posting the same on social media would receive a quicker response as well as the count of citations tends to increase rapidly every time it is referred, receiving immediate attention.

Based on the state-of-the-art for the scholarly use of social media [4], we narrow down the identification of the engaging groups in scholarly communication. Observe the generalization of findings regarding altmetrics based on feature selection, the use of social media in academia and various ways that are used to cite the article in the social media. YouTube being our consideration, we would also be able to categorize more on other features such as number of views, likes and dislikes based on the features like category or subject from the YouTube dataset in our future predictions after performing the data collection for the same.

Measuring the rate at which the contents are posted on the social media based on the previously collected information helps us in estimating the cost and accurately predicting its popularity [5]. The evaluation of the model was checked for the 10-fold cross validation by randomly dividing each dataset into training and test set. Analysis on the performance of the various model likes Szabo-Huberman model, Multivariate Linear model, etc. is detailed to compare the best performing one for YouTube Category. By assigning different weights to different popularity samples, the model would be able to differentiate videos on popularity evolution pattern and helps in reducing the errors on the predicted set.

3 Dataset

Altmetrics dataset [6] was chosen for this project. The reduced altmetrics dataset worked on had about 100,000 tuples from which the analysis was done. This helped the understanding the qualitative data that gives insights into the citation-based metrics [1]. Altmetrics dataset contains normalized qualitative information about any given research article from which we are taking into account a subset of data which is used for categorizing based on the URL defined in one of the features of the dataset. The features were validated against the YouTube label which holds the value as a 0 or 1 depending on whether the paper will be posted on YouTube or not. The features chosen were 'Altmetric_ID', 'Altmetric_Score', 'Mendeley', 'CiteULike', 'Twitter', 'Facebook', Video', 'GooglePlus', 'Reddit', 'Blogs', 'Peer_Reviews', 'News', 'F1000', 'Wikipedia', 'Youtube'. Moving forward, the entire Altmetrics dataset had about 380,000 tuples, which was used for the further analysis and for performing the necessary operations.

'Youtube' was taken as the class label whose values are 0 or 1 depending on whether the Altmetric_ID has the keyword youtube under posts -> video -> url. Features mendeley and citeulike are under counts-> readers. The remaining features are taken from counts -> respective feature -> posts_count.

The Altmetric_Score [21], being the weighted count of attention that the citation received on the various platforms, was dropped due to having a high correlation with the rest of the features chosen. It came to our attention that Video was also highly correlated with that of the target, which was removed as well.

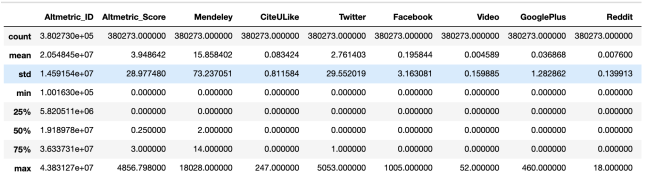
4 Data Cleaning

Once the data is populated into the dataframe of a total of 15 data columns. Altmetrics_Score was dropped from this and the datatype was converted to integer format from float for all the column values using downcast. We noticed that 'Mendeley' column had higher range of values. Normalization [10] was performed using StandardScaler from the Scikit learn library [9] in order to overcome this. The result of this was that all the data was brought to the same range.

We also came across many NaN value which was dropped using the dropna function from the pandas dataframe. This helped us in analyzing and dropping the rows or columns with null values.

4.1 Descriptive Statistics

The analysis on a data can be done by describing or summarizing the data in a meaningful way from which any patterns if present can be deciphered [38]. Further this aid in building the majority component of concluding the quantitative data analysis. It was given to us that post counts are the number of unique items created by those users that mention this article. The same user could blog, tweet or otherwise share the same article more than once, so posts_count can be greater than unique_users_count from the feature list.



	Altmetric_ID	Altmetric_Score	Mendeley	CiteULike	Twitter	Facebook	Video	GooglePlus	Reddit
count	3.802730e+05	380273.000000	380273.000000	380273.000000	380273.000000	380273.000000	380273.000000	380273.000000	380273.000000
mean	2.054845e+07	3.948642	15.858402	0.083424	2.761403	0.195844	0.004589	0.036868	0.007600
std	1.459154e+07	28.977480	73.237051	0.811584	29.552019	3.163081	0.159885	1.282862	0.139913
min	1.001630e+05	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	5.820511e+06	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	1.918978e+07	0.250000	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	3.633731e+07	3.000000	14.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000
max	4.383127e+07	4856.798000	18028.000000	247.000000	5053.000000	1005.000000	52.000000	460.000000	18.000000

Figure 1: Descriptive statistics

5 Feature Selection

Selecting features manually was an important part of emphasizing which feature contributes more towards in prediction of the output [37]. Improper feature selection leads to the decrease in the accuracy values and misleading to understand the concepts of overfitting and underfitting based on the results. Some of the selection techniques are Univariate Selection, Feature Importance and Correlation matrix with heatmap.

5.1 Correlation Matrix

The correlation matrix [12] [13] tells us how our features and target variables are related to one another by giving the correlation coefficient between them. They can increase or decrease the value of the target variable with the increase in the value of the features giving us an insight into in depth analysis. The correlation matrix was plotted using heatmap by importing the seaborn and matplotib [11] for 2D plotting. The annot value was set as True to display the values in each cell and cmap helps us to map the data values to the color space.

From the pattern obtained visually, Altmetrics_Score was highly correlated with other columns since it records the count of the attention it receives across the social media, so this column was dropped. When logistic regression model was applied on the dataset with video column, we obtained 100% accuracy. This made us to reconsider the points such as overfitting while we noticed that the video feature, whose value count is 1111 was highly correlated with that of YouTube column values. So, we dropped video column, giving us a total of 11 features to focus on for our predictions.

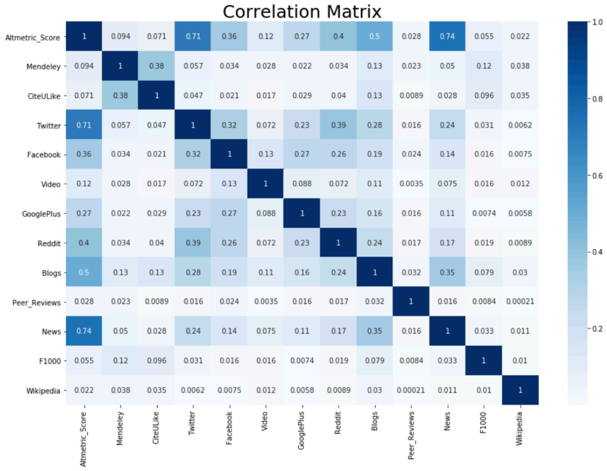


Figure 2: Correlation Matrix with YouTube as target variable

6 Classification

Four classification models were chosen and each of it is individually explained below. For splitting the dataset, train_test_split [22] was used from sklearn.model_selection. The ratio of the split is randomized with 70% for training and 30% for testing on the unseen values. The goal behind the train test split is, we need to train our model on the known output from which the model learns the data. This can be used to generalize on the unseen data. The concept of regularization which is used to prevent overfitting of the training data with the hope that it improves the generalization, i.e., the ability to correctly handle the data that the network has not trained on. Stratify parameter was set to yes in order to return the same proportions of the class labels as the input set from the training and test subsets.

6.1 Logistic Regression

Logistic regression [14] predicts the binary probabilities by giving the measure of the relationship between the dependent categorical variable and the independent variables. The probability of the event obtained is represented as a linear function of a combination of predicted variables and the independent variables should not be correlated. They have low variance and high bias and the probability of outcome works well for decision boundaries [15].

We imported the LogisticRegression from sklearn.linear_model. We created the instance of the logistic regression and fit the model using the training data. Prediction was performed on the training results to check if the training and the testing accuracy was close to each other to prevent the model from overfitting.

Taking equal samples of the class labels for citation present in YouTube and not present in YouTube was considered for the logistic regression with the train test split as 70- 30. Which also gave us an accuracy of 100%. But when the same model was applied on the dataset after removing the video feature, the accuracy was still the same and it was also getting some false negative values [16] this time. However, the precision and recall dropped its value for class 1 in comparison to having high values before.

Class imbalance problem was handled by applying the techniques such as oversampling, undersampling, SMOTE. Amongst which Logistic regression with SMOTE and oversampling was performing the best in terms of accuracy and as a model in comparison with the other ones.

6.1.1 Logistic Regression with Oversampling

The logistic regression with oversampling was performed to a highly imbalanced dataset gave us a better model. This was done by importing the resample from sklearn.utils. The oversampled class values were 265413 for both the classes. The accuracy and recall for oversampled data are 92% and 0.55 which can be shown by classification report.

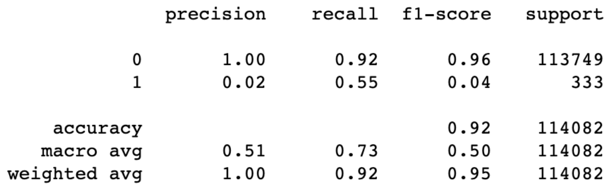

```
              precision    recall  f1-score   support

           0       1.00      0.92      0.96    113749
           1       0.02      0.55      0.04       333

    accuracy                           0.92    114082
   macro avg       0.51      0.73      0.50    114082
weighted avg       1.00      0.92      0.95    114082
```

Figure 3: Classification Report- LR with oversampling

6.1.2 Logistic Regression with SMOTE

The logistic regression with smote provided us with an accuracy of 90% and recall value for the same for 0.60 after synthetic data generation for handling the imbalanced data.

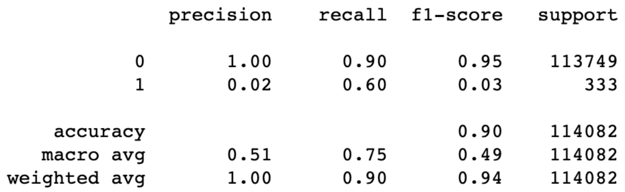

```
              precision    recall  f1-score   support

           0       1.00      0.90      0.95    113749
           1       0.02      0.60      0.03       333

    accuracy                           0.90    114082
   macro avg       0.51      0.75      0.49    114082
weighted avg       1.00      0.90      0.94    114082
```

Figure 4: Classification Report- LR with SMOTE

6.2 Random Forest

As the name depicts, random forest classifier comprises of large number of individual decision tree algorithm. The random forest is a classification algorithm which falls under the category of supervised learning model and is a combination of several decision trees that uses ensemble [18]. This model can be implemented for both classification and regression problems. Every individual tree in the random forest gives out its prediction and the votes are calculated. The class with maximum number of votes are chosen to be the final prediction. For splitting the node, this model considers a random subset of the feature. Instead of searching for the best threshold, this model uses the random threshold for producing more trees. and model is very convenient to measure the significance of each prediction. The model provides the better result in the short amount of time.

The class imbalance problem was handled using SMOTE for the random forest classification model. The prediction showed an accuracy of 96% but we did not find it in our favor because of the number of positive class i.e., class 1 values was as low as 19 for true negative with the false positive [17] having the count of 314 which corresponds to the class 0 values extracted.

6.3 Decision Tree

As the name depicts, Decision tree algorithm follows the structure of the tree where each branch represents the decision rule and the leaves represent the outcome. They are constructed on a top to bottom fashion in a divide and conquer manner. Each partition on this rule occurs based on the attribute value. The partition occurs in a recursive manner. Since this algorithm shares the internal decision-making logic it falls under the category of white box type algorithm in Machine Learning algorithm [34]. The time it takes to execute is faster when compared to black box type algorithms such as neural networks. The time it takes to train the model is based on the number of records and the number of attributes in the data set. This algorithm does not rely on the probability-based assumptions. The results are predicted with high accuracy despite of handling high dimensional dataset [40].

We used entropy as our criterion value which is the measure of uncertainty associated with a random number. Due to having a highly imbalanced dataset, over sampling and undersampling was chosen to perform on a decision tree classifier. The result of the oversampled accuracy was about 92% but upon plotting a confusion matrix, we noticed that a greater number of false positives was getting classifies than the true negative values. Also, the quite a large number of false negatives was obtained which is why we did not consider this model. In the case of decision tree on undersampling led to a high number of false negative values which is the case of having the not spam mails getting classified under spam making [20] reducing its reliability.

6.4 Support Vector Machine

The SVM uses a hyperplane on a N-dimensional space to classify the datapoints. The goal behind this is to maximize the margin of the classifier. And the algorithm is implemented using kernels. The SVC [19] (support vector classifier) uses a radial basis function kernel. It takes the input data and converts it to a resulting format. SVM performs well with a nonlinear boundary depending on the kernel chosen. They also handle the high dimensional data well.

This model ran for under sampled and oversampled data giving the lowest accuracy in comparison with the other models of about 88% and also having a very low recall value.

7 Class Imbalance

In the real-world classification of any given dataset, the classes have a high chance of being imbalance leading to a class imbalance problem which is not a cost sensitive implementation of all the learning algorithms. This means that the classes are not equal in number. One the class value is more than the other and there are several techniques to resolve this issue which would be discussed later.

Several research papers and solutions to similar problems in the past suggest that a class imbalance problem [34] can be resolved using either oversampling the minority class or under sampling the majority class. The ideal goal behind the need to process the imbalance data before they are feed into the classifier is to overcome the property of the classifier being more sensitive to the majority class and less sensitive to the minority class. This would result in the prediction of the majority class. The limitations of oversampling a minority class in an imbalanced dataset leads to overfitting since it copies the existing values [7]. The concerns with undersampling is that is misses out the essential data.

In our case, class 0 was 379,162 while class 1 had only 1111 values.

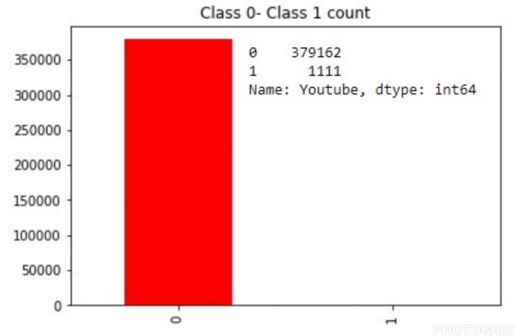


Figure 5: Bar chart to show Class Imbalance problem

7.1 Oversampling & Undersampling

For a case of highly imbalanced dataset, there is a high possibility that the model shows bias towards the majority class and ignore the minority class. This can be overcome by sampling the data in a way that the model shows no bias [39]. We applied both oversampling and under sampling techniques to our data to verify the performance. In the case of oversampling the minority class samples are increased to be equal to the majority class samples. In the case of under sampling the majority class samples are down sampled to be equal to the minority class samples. These are done using resample from scikit-learn library. Logistic Regression model is applied on both the oversampled and under-sampled data each having an accuracy of about 92% and 90% respectively.

7.2 Smote

SMOTE is another way to deal with imbalanced data. SMOTE generates synthetic samples to balance the data. minority classes are up sampled, and majority classes are down sampled with these synthetic samples [31]. After applying SMOTE, the samples of both the classes is equal to 265413. We applied Logistic Regression on SMOTE data and the results were almost similar to the LR with oversampled data. The accuracy is 90% whereas recall is equal to 0.60. After verifying these results, we decided on applying SMOTE data to another model Random Forest Classifier. This model resulted in much less recall and very few true negatives.

7.3 XGBoost

Extreme gradient boosting is a supervised machine learning technique for supporting various objective functions, in solving the classification and regression problems. It is built on the gradient boosting framework [24] to work with the extreme computation limits to provide the scalable, portable and accurate library. They are. A more regularized model to control overfitting resulting in better performance. It works better when we have lot of training data where the feature is a mix of categorical and numeric values. However, it doesn't perform well in case of categorical and when the row values hugely differ with the feature values.

8 Classification Metrics

The most commonly used approach to solve any given problem in machine learning is classification [28]. The above-mentioned models were applied.

Classification report calculates the precision, recall, f1 score and accuracy. Precision and recall values can simply not be used as the only measure to indicate the performance of any model. We need more than just precision and recall concluding the best resulting model. In a class imbalanced problem, the accuracy would be biased more towards the most frequent class. In order to overcome this, we used the class specific metrics like precision. In our case for the articles being cited in YouTube or not, the calculation for this is given by,

Precision = True Positive / (True Positive + False Positive)

Similarly, Recall is also used to define the fraction of samples that are correctly predicted by the model. They are defined as,

Recall = True Positive / (True Positive + False Negative)

F1- Score is the harmonic mean score instead of an arithmetic mean between the precision and recall as it punishes the extreme values the most. It is given as,

F- Measure = (2 * Recall * Precision) / (Recall + Precision)

The accuracy rate gives the percentage based on which we can categorize and arrive at a conclusion if our model performs better or not.

Accuracy = (True Positive + True Negative) / (True Positive + True Negative + False Positive + False Negative)

Based on all of the above calculation, the model that performed well are Logistic Regression for oversampled dataset and Logistic Regression using SMOTE.

8.1 Confusion Matrix

The evaluation of the performance of the classification model can be better understood visually by a confusion matrix [27]. Ours is a binary classification, making the prediction belonging to the positive or the negative class. The confusion matrix is divided into four categories- True Positive (TP) indicating the observed and the predicted values are to be positive , True Negative (TN) indicated that the observed and the predicted values are to be negative, False Positive (FP) indicates that the observation is negative and the predicted values is positive and False Negative (FN) indicated that observed value is positive and the prediction is negative. The observations for positive in our case are that the paper is cited in YouTube and the observation for negative is that article is not present in YouTube.

The below confusion matrix is the result of two of the best performing models,

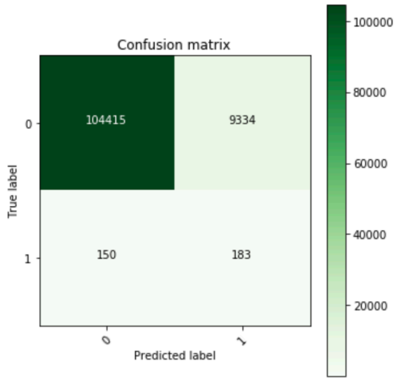


Figure 6: Confusion matrix for Logistic Regression after Oversampling

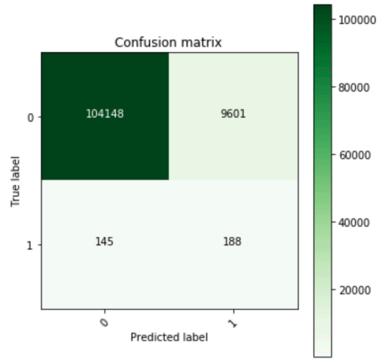


Figure 7: Confusion metrics for Logistic Regression with SMOTE

8.2 ROC Curve

Receiver operating characteristic curve is a plot that illustrates the diagnostic for binary classifier and its discrimination threshold is varied. The Roc curve is plotted with the True Positive Rate (TPR) on the y-axis and the False Positive Rate (FPR) on the x-axis [30]. TPR is the fraction of positive samples that are correctly labeled. FPR is the fraction of misclassified negative samples as positive samples.

```
AUC: 0.85
```

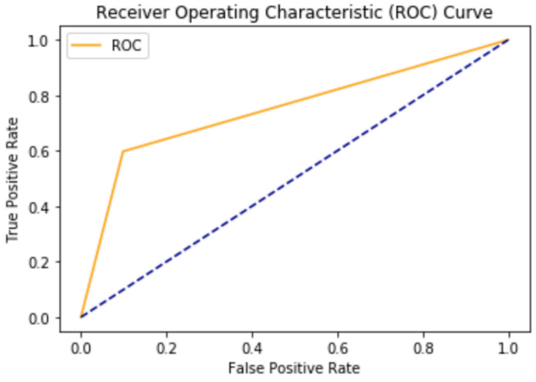


Figure 8: ROC Curve

8.3 Precision Recall curve

The precision and recall curves show a tradeoff between the recall and precision for different thresholds [29]. We do not use the TN values for making a curve. The below curve illustrates the plotting of the data points for precision and recall,

```
Logistic: f1=0.037 auc=0.073
```

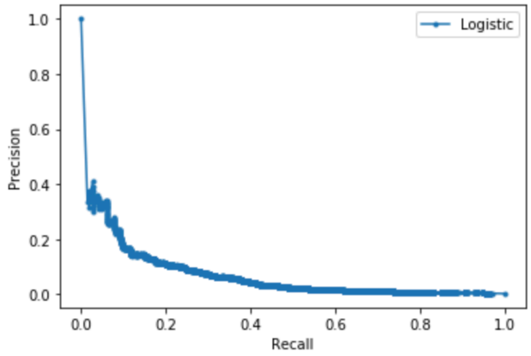


Figure 9: Precision Recall Curve

9 Overfitting

Although the dataset taken into consideration had a reasonable number of tuples to consider for the experiment, the number of class 1 values i.e., the videos getting posted on YouTube was low leading to an imbalanced dataset. Performance of any model thrown at it was producing high accuracy. So, we tested for the overfitting condition [23] and ways to avoid such situations encountered. Considering equal samples of positive and negative classes we taken and applied the Logistic Regression model. Also, applied the LR model for the prediction of accuracy of training set of results with the testing accuracy and noticed that the result was not overfitting, just that out model has pretty high accuracy from the start for any classification model.

10 Conclusion

The Model that performs better with regards to the accuracy, precision and recall values along with the number of observed and predicted values matching Logistic regression was the best. The class imbalanced problem was also handled and the preferred choice depending on our results for classifying is. The oversampled and SMOTE.

11 Future Work

Our future work includes in trying with another recent dataset from the altmetrics to check how the precision and recall varies in accordance. Implementation of more models like Naïve Bayes and doing a regression for predicted of the number of likes and dislikes the articles has received on YouTube.

REFERENCES

[1] Bornmann, Lutz. 2014. "*Do Altmetrics Point to the Broader Impact of Research? An Overview of Benefits and Disadvantages of Altmetrics*." Journal of Informetrics 8 (4): 895–903.

[2] Weller, Katrin. (2015). Social Media and Altmetrics: *An Overview of Current Alternative Approaches to Measuring Scholarly Impact*. 10.1007/978-3-319-09785-5_16.

[3] Thelwall, Mike & Haustein, Stefanie & Larivière, Vincent & Sugimoto, Cassidy. (2013). *Do Altmetrics Work? Twitter and Ten Other Social Web Services*. PloS one. 8. e64841. 10.1371/journal.pone.0064841.

[4] Sugimoto, Cassidy & Work, Sam & Larivière, Vincent & Haustein, Stefanie. (2016). *Scholarly use of social media and altmetrics: A review of the literature*. Journal of the Association for Information Science and Technology. 10.1002/asi.23833.

[5] Pinto, Henrique, Jussara M. Almeida, and Marcos A. Gonçalves. "Using early view patterns to predict the popularity of youtube videos." *In Proceedings of the sixth ACM international conference on Web search and data mining, pp. 365-374. ACM, 2013.*

[6] https://blog.une.edu.au/library/2019/02/20/altmetric-com-on-site/

[7] Weiss, Gary M., Kate McCarthy, and Bibi Zabar. "Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs?." Dmin 7, no. 35-41 (2007): 24.

[8] Liaw, Andy, and Matthew Wiener. "Classification and regression by randomForest." *R news* 2, no. 3 (2002): 18-22.

[9] Garreta, Raúl, and Guillermo Moncecchi. *Learning scikit-learn: machine learning in python*. Packt Publishing Ltd, 2013.

[10] Buitinck, Lars, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae et al. "API design for machine learning software: experiences from the scikit-learn project." *arXiv preprint arXiv:1309.0238* (2013).

[11] Bisong, Ekaba. "Matplotlib and Seaborn." In *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, pp. 151-165. Apress, Berkeley, CA, 2019.

[12] Dziuban, Charles D., and Edwin C. Shirkey. "When is a correlation matrix appropriate for factor analysis? Some decision rules." *Psychological bulletin* 81, no. 6 (1974): 358.

[13] Steiger, James H. "Tests for comparing elements of a correlation matrix." *Psychological bulletin* 87, no. 2 (1980): 245.

[14] Dreiseitl, Stephan, and Lucila Ohno-Machado. "Logistic regression and artificial neural network classification models: a methodology review." *Journal of biomedical informatics* 35, no. 5-6 (2002): 352-359.

[15] Lemon, Stephenie C., Jason Roy, Melissa A. Clark, Peter D. Friedmann, and William Rakowski. "Classification and regression tree analysis in public health: methodological review and comparison with logistic regression." *Annals of behavioral medicine* 26, no. 3 (2003): 172-181.

[16] Beguería, Santiago. "Validation and evaluation of predictive models in hazard assessment and risk management." *Natural Hazards* 37, no. 3 (2006): 315-329.

[17] Griffin, Gregory, Alex Holub, and Pietro Perona. "Caltech-256 object category dataset." (2007).

[18] Liaw, Andy, and Matthew Wiener. "Classification and regression by randomForest." *R news* 2, no. 3 (2002): 18-22.

[19] Teli, Savita Pundalik, and Santosh Kumar Biradar. "Effective email classification for spam and non-spam." *International Journal of Advanced Research in Computer and software Engineering* 4, no. 2014 (2014).

[20] Sharma, Aman Kumar, and Suruchi Sahni. "A comparative study of classification algorithms for spam email data analysis." *International Journal on Computer Science and Engineering* 3, no. 5 (2011): 1890-1895.

[21] Thelwall, Mike, and Tamara Nevill. "Could scientists use Altmetric. com scores to predict longer term citation counts?." *Journal of Informetrics* 12, no. 1 (2018): 237-248.

[22] Godbole, Shantanu, and Sunita Sarawagi. "Discriminative methods for multi-labeled classification." In *Pacific-Asia conference on knowledge discovery and data mining*, pp. 22-30. Springer, Berlin, Heidelberg, 2004.

[23] Jabbar, H., and D. R. Z. Khan. "Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study)." *Computer Science, Communication and Instrumentation Devices* (2015).

[24] Chen, Tianqi, Tong He, Michael Benesty, Vadim Khotilovich, and Yuan Tang. "Xgboost: extreme gradient boosting." *R package version 0.4-2* (2015): 1-4.

[25] Chaturvedi, K. K., and V. B. Singh. "Determining bug severity using machine learning techniques." In *2012 CSI Sixth International Conference on Software Engineering (CONSEG)*, pp. 1-6. IEEE, 2012.

[26] Davis, J., & Goadrich, M. (2006, June). The relationship between Precision-Recall and ROC curves. *In Proceedings of the 23rd international conference on Machine learning (pp. 233-240). ACM.*

[27] Johnson, Amos Y., and Aaron F. Bobick. "Relationship between identification metrics: Expected confusion and area under a roc curve." In *Object recognition supported by user interaction for service robots*, vol. 3, pp. 662-666. IEEE, 2002.

[28] Melnyk, Steven A., Douglas M. Stewart, and Morgan Swink. "Metrics and performance measurement in operations management: dealing with the metrics maze." *Journal of operations management* 22, no. 3 (2004): 209-218.

[29] Buckland, Michael, and Fredric Gey. "The relationship between recall and precision." *Journal of the American society for information science* 45, no. 1 (1994): 12-19.

[30] McClish, Donna Katzman. "Analyzing a portion of the ROC curve." *Medical Decision Making* 9, no. 3 (1989): 190-195.

[31] Chawla, Nitesh V., Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. "SMOTE: synthetic minority over-sampling technique." *Journal of artificial intelligence research* 16 (2002): 321-357.

[32] Forman, George. "An extensive empirical study of feature selection metrics for text classification." *Journal of machine learning research* 3, no. Mar (2003): 1289-1305.

[33] Weston, Jason, Sayan Mukherjee, Olivier Chapelle, Massimiliano Pontil, Tomaso Poggio, and Vladimir Vapnik. "Feature selection for SVMs." In *Advances in neural information processing systems*, pp. 668-674. 2001.

[34] Kubat, Miroslav, and Stan Matwin. "Addressing the curse of imbalanced training sets: one-sided selection." In *Icml*, vol. 97, pp. 179-186. 1997.

[35] Picek, Stjepan, Annelie Heuser, Alan Jovic, Shivam Bhasin, and Francesco Regazzoni. "The curse of class imbalance and conflicting metrics with machine learning for side-channel evaluations." (2018).

[36] Jacobsen, Carsten, Uwe Zscherpel, and Petra Perner. "A comparison between neural networks and decision trees." In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pp. 144-158. Springer, Berlin, Heidelberg, 1999.

[37] Dash, Manoranjan, and Huan Liu. "Feature selection for classification." *Intelligent data analysis* 1, no. 1-4 (1997): 131-156.

[38] Oja, Hannu. "Descriptive statistics for multivariate distributions." *Statistics & Probability Letters* 1, no. 6 (1983): 327-332.

[39] Yap, Bee Wah, Khatijahhusna Abd Rani, Hezlin Aryani Abd Rahman, Simon Fong, Zuraida Khairudin, and Nik Nik Abdullah. "An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets." In *Proceedings of the first international conference on advanced data and information engineering (DaEng-2013)*, pp. 13-22. Springer, Singapore, 2014.

[40] Dietterich, Thomas G. "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization." *Machine learning* 40, no. 2 (2000): 139-157.