Help Boost Our Online Reach

(Team Name : cloud9)

Vishwajeet Deulkar
IIITB (MT2021154)
vishwajeet.deulkar@iiitb.ac.in

Manan Mehta
IIITB (MT2021517)
manan.mehta@iiitb.ac.in

Abstract - The ad-agency spends a considerable amount of time and money to find the best web pages to publish their ads on. The aim of this task is to identify the relevant, high-quality web pages from a pool of user-curated web pages, for the identification of "ad-worthy" web pages. The challenge requires building large-scale, end-to-end machine learning models that can classify a website as either "relevant" or "irrelevant", based on attributes such as alchemy category and its score, meta-information of the web pages and a one-line description of the content of each webpage.

Index Terms -: Introduction, Data Preprocessing on train-test data by merging (Dealing With Missing Values, One Hot Encoding, outlier detection), NLP preprocessing on text data columns (lowering case, punctuation marks removal, hyperlink removal, stripping, stop words removal, stemming, tokenization), using BOW and TF-IDF for vectorizing text data in one solution, using GloVe for vectorizing text data in other solution, Train test split, Applying model and predicting output probabilities, Stacking of models, Conclusions, References.

I. INTRODUCTION

Today brands are spending a considerable amount of money for marketing and branding and the major way of this is by advertisement. They are approaching online advertising agencies for ads on online mode. They select web pages that will generate prolonged online traffic so that their ads can have a long-lasting reach.
The goal of this project is to, to predict which web pages would attract high, long-lasting traffic and to identify "ad-worthy" web pages.

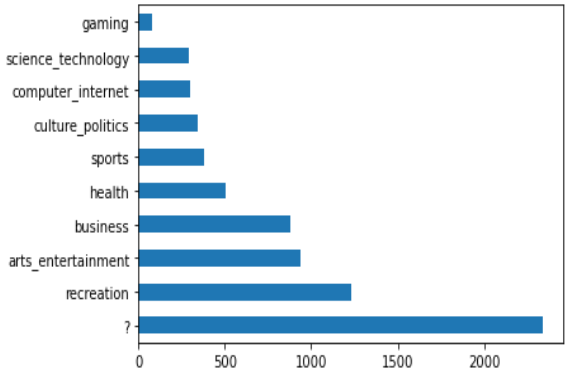
II. DATASET

Training dataset i.e. train_data.csv has size of (5916,27) and testing dataset i.e test_data.csv has size of (1479, 26). We have to predict the missing column in test data, which is the label. After predicting output value, we should have some way to classify a website as either "relevant" or "irrelevant".
For kaggle competition, we have to use the Area Under the Receiver Operating Characteristic Curve (ROC AUC) score as the classification metric.

III. EXPLORATORY DATA ANALYSIS

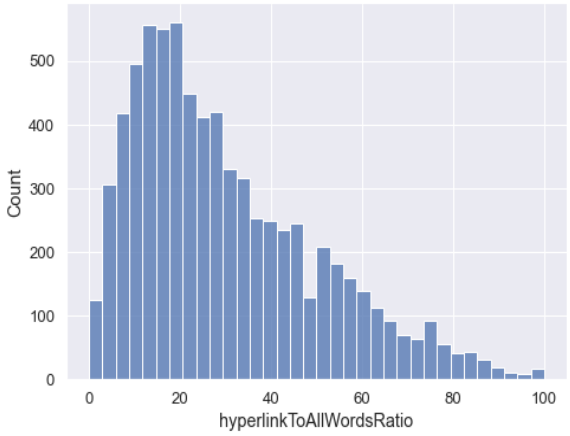
It is important to know the distribution of various features along data to get insights during feature selection and preprocessing of data.It helps to drop data which was recorded by errors in any observations. Few of which are represented below:

1) Distribution of “alchemey_category” feature:

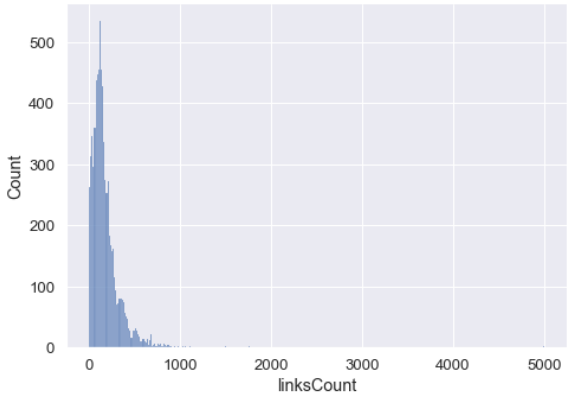


For most rows “alchemy_category” has null(?) missing values which are replaced by other values.

2) Distribution of hyperlinkToAllWordsRatio over the data:



3) Distribution of LinksCount over the data:



IV. DATA PREPROCESSING AND FEATURE EXTRACTION

A. DEALING WITH MISSING/NULL VALUES

In our dataset we have missing values for four columns "alchemy_category", “isNews ”, "alchemy_category_score"and “isFrontPageNews” respectively. We replace missing values of "alchemy_category" by random choice of other values. For the numerical column "alchemy_category_score" we replace values with mean. For categorical columns “isNews”, “isFrontPageNews” we replace missing values with respective modes. Column “framebased” has all 0 values hence the column was dropped.

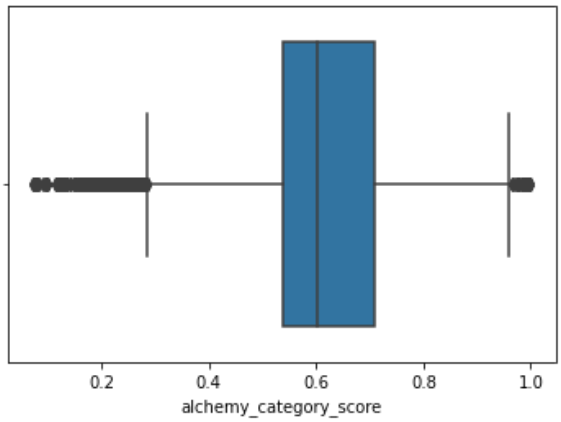
B. ONE HOT ENCODING

We have used one-hot encoding to handle data in a categorical column named “alchemy_category”.

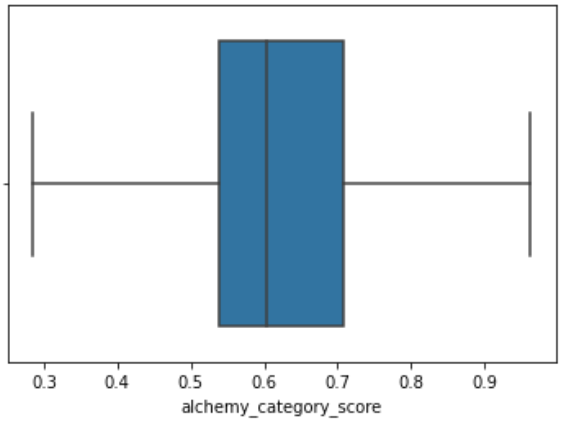
C. OUTLIER DETECTION AND REMOVAL

IQR is a good statistic for summarizing a non-Gaussian distribution sample of data; we used this method for outlier detection and removal. Few examples are shown below:

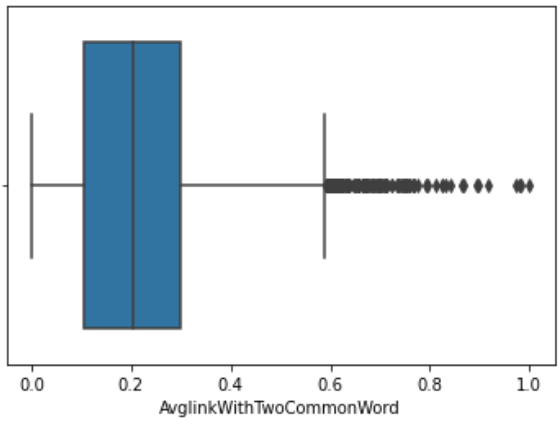
1) alchemey_category_score distribution :



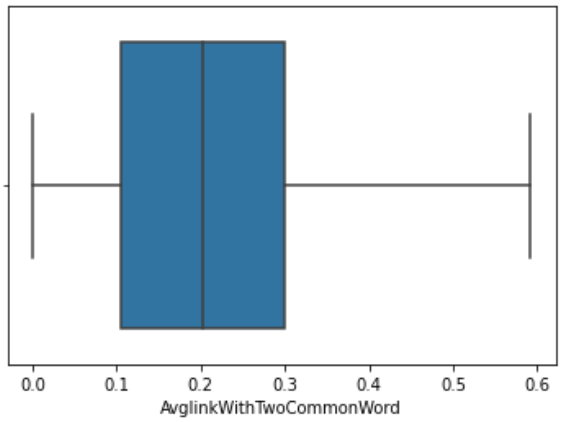
After removing outliers :



2) AvglinkWithTwoCommonWord distribution :



After removing outliers:



D. NORMALIZATION

We have done normalization for rescaling feature values in the range[0,1] using robust scaling.

IV. NLP DATA PREPROCESSING

Hyperlink data of the “URL” column has been removed. The main text column to apply text preprocessing is “webdescription” we will remove title,body and URL tags from the rows of the column.

Basic text pre-processing on “webdescription" column:

1) Converting all strings to lowercase.
2) Removing any special characters or hyperlink related characters.
3) Removing stop words, stripping no-alphabetic characters.
4) Applying lemmatization to find out keywords.

To train data we need to convert raw data into more sophisticated data. This means we need to convert text data into numerical data to get insights from data to train models. To convert words into numerical values to be meaningful for machine learning algorithms and models, we use algorithms such as One Hot Encoding, Bag Of Words, TF-IDF, WordVec, GloveVec.

CONVERTING TEXT FEATURES TO NUMERICAL

1) APPLY BAG OF WORDS FOR TEXT FEATURES

For computation on machine learning models we need to convert text data into numerical data as they won't understand raw-text data. We can convert text column "webdescription" into numerical by using Bag Of Words technique. It can be seen as one-hot encoding for each word in rows of data.

2) APPLY TF-IDF WORD EMBEDDING

TF-IDF assigns value to a word in a document according to its importance divided by its importance across all documents in the corpus. We can write this in mathematical form as :

$$TF(d_i^j) = \frac{frequency\ d_i^j\ appears\ in\ d_i}{\#\ of\ words\ in\ d_i}$$

$$IDF(w) = log(\frac{\#\ documents\ in\ corpus}{\#\ documents\ that\ contains\ word\ w}$$

$$TF - IDF = TF \times IDF$$

Output is a sparse matrix as the feature will get significant value only if it is present in that name or description. We have encoded webdescription feature using TF-IDF.

3) APPLYING GLOVE EMBEDDING

It is used for vectorizing words in text documents. GloVe is an unsupervised learning algorithm in which training is performed on aggregated global word-word co-occurrence statistics.

After getting output from all these steps, models are applied as shown below.

V. MODEL IMPLEMENTATION

After doing all the EDA, Data preprocessing and feature extraction(Dealing with missing values,One Hot encoding, Normalization, Outlier detection) and NLP text data preprocessing, we have experimented training data with the following models.

Models with TF-IDF as text data preprocessor

1) Logistic Regression
2) Multilayer Perceptron Classifier
3) SVM
4) Random forest
5) Simple Xgboost classifier

Models with GloVe as text data preprocessor

6) Logistic Regression
7) Multilayer Perceptron Classifier
8) SVM RBF
9) Random forest
10) Stacking

1. LOGISTIC REGRESSION

Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. As our target variable is binary, we used logistic regression as a binary classifier. We have done hyper parameter tuning using the max_iter parameter.

2. MULTILAYER PERCEPTRON CLASSIFIER

It is a type of Artificial Neural Network, it contains at least three layers of node input layer, hidden layer and an output layer. We have done hyper parameter tuning using alpha and max_iter parameters.

3. SUPPORT VECTOR MACHINE

SVM is a supervised learning algorithm which is used for classification. The goal of SVM is to find a hyperplane in N-dimensional space that can classify all data points distinctly.

4. RANDOM FOREST

Random forest is used to solve classification and regression problems; it uses ensemble learning that combines many classifiers to provide solutions to complex problems. It consists of many decision trees.

5. SVM RBF

We used the RBF(radial basis function) kernel as it helps SVM for a hyperplane to become nonlinear.

6. XGBOOST

It is a system of tree boosting which is a highly effective, flexible,portable and widely used machine learning method. It implements a machine learning algorithm under gradient boosting technique.

7. **Stacking**

Stacking allows users to reap the benefits of different models by combining their individual predictions using higher levels. We used SVM RBF and MLP in base models and logistic regression as the final(meta) model as of now.

VI. TABLES OF MODELS USED AND RESPECTIVE SCORES

Preprocessing	Model	Public Score on Kaggle	Private score on Kaggle
Dealing With Missing Values, One Hot Encoding, NLP data preprocessing, BOW vectorizer	logistic regression	0.79160	0.80789
Dealing With Missing Values, One Hot Encoding, NLP data preprocessing, TF IDF vectorizer	logistic regression	0.87392	0.88947
Dealing With Missing Values, One Hot Encoding, NLP data preprocessing, TF IDF vectorizer	Multi layer Perceptron classifier	0.82188	0.84003
Dealing With Missing Values, One Hot Encoding, NLP data preprocessing, TF IDF vectorizer	SVM	0.86601	0.87988

Table(1) : Implementing model using TF-IDF vectorizer

Preprocessing	Model	Public Score on Kaggle	Private score on Kaggle
Dealing With Missing Values, One Hot Encoding, NLP data preprocessing, GloVe Embedding	logistic regression	0.892221	0.87776
Dealing With Missing Values, One Hot Encoding, NLP data preprocessing, GloVe Embedding	SVM RBF	0.88928	0.87095
Dealing With Missing Values, One Hot Encoding, NLP data preprocessing, GloVe Embedding	Multi layer Perceptron classifier	0.89238	0.87382
Dealing With Missing Values, One Hot Encoding, NLP data preprocessing, GloVe Embedding	Random Forest	0.86086	0.83794
Dealing With Missing Values, One Hot Encoding, NLP data preprocessing, GloVe Embedding	Stacking	0.89536	0.87912

Table(2) : Implementing model using GloVe Embedding

VII. INDIVIDUAL CONTRIBUTIONS

Vishwajeet Deulkar	EDA, Cleaning of data, TF-IDF, BagOfWords, Random Forest, Stacking, Logistic Regression, SVM Implementation
Manan Mehta	Preprocessing, Analysis of Columns, MLP and GloVe implementation, Hyperparameter Tuning

VIII. CONCLUSION

It was a great learning experience in learning new things like NLP. Overall it was a fun ride. We would like to conclude that we were able to come

up with an efficient model to predict “ad-worthy” websites or not. We got a private score of 0.88947.

IX. ACKNOWLEDGMENT

We would like to thank Professor G. Dinesh Babu Jayagopi , Dr. Neelam Sinha, our project mentor Sarthak Khoche, Ayush Yadav and our subject Teaching Assistant Swasti Mishra for giving us the insights in ML field and helping us whenever we were stuck by giving us ideas and resources to learn from.

X. REFERENCES

[1] **Simple Model Stacking, Explained and Automated**

[2] **Your Guide to Natural Language Processing (NLP)**

[3] **Normalized Nerd YouTube channel for learning basics of NLP**

[4] **Machine Learning Classifiers**

[5] **Feature Engineering – Detect and Remove Outliers**