Website Classification Using Machine Learning Approaches

The Bachelor's thesis

Domantas Meidus

Vilnius University
Faculty of Mathematics and Informatics
Institute of Computer Science
Department of Computational and Data Modeling

2019

Outline

1. Data set
2. Data preprocessing
   - Website scrapping and text parsing
   - Word Tokenization
   - Word Lemmatization
   - Stop words
   - Website language Determination
   - Most frequent words list for each category
3. Machine Learning
   - Features and Labels creation
   - Custom Model
   - Machine Learning Models
4. Results

Data set (1)

Open source Data Set

- Data set is created in 2015
- Data set is open source: https://www.figure-eight.com/data-for-everyone/
- Data set contains 31086 websites with 25 different categories
- Categories list:

Adult	Finance	News & Media
Arts & Entertainment	Food & Drink	People & Society
Automotive	Gambling	Pets & Animals
Beauty & Fitness	Games	Reference
Books & Literature	Health	Science
Business & Industry	Home & Garden	Shopping
Career & Education	Internet & Telecom	Sports
Computer & Electronics	Law & Government	Travel

Data set (2)

Custom Data Set

- Data set is custom made.
- Website categories are validate by human.
- Data set contains 282 websites with 25 different categories

Data preprocessing

1. Website scrapping and text parsing
2. Word Tokenization
3. Word lemmatization
4. Stop Words
5. Website language Determination
6. Most frequent words list for categories

Website scrapping and text parsing

- Website content is downloaded by sending GET request to website via **urlib.request** Python library
- **CSS** and **JS** HTML tags are excluded
- Website HTML code is parsed into raw text by using **BeautifulSoup** Python library

Word Tokenization

Word Tokenization is a process of splitting up a larger body of text into smaller words.

Example of word tokenization

Normal text: 'The First sentence is about Python.'
Tokenized words: ['The','First', 'sentence', 'is', 'about', 'Python', '.']

Word Lemmatizations

Word Lemmatization is the morphological analysis of the words.

Word Lemmatization is done by using **nltk WordNetLemmatizer** Python library

Example of Word Lemmatization

Word	Lemma
Studies	Study
Studying	Study
Children	Child
Study	Study

Stop words

Stop words are considered to be common words in a particular language which do not provide lots of information by using itself.

- English Stop Words list is imported by using **nltk** Python library.
- Library contains 179 english language stop words.

Website language Determination

Process of website language determination:

1. Exclude all website words that are not in **nltk.corpus.words.words** library english language words vocabulary.
2. Calculate percentage of english words in a website
3. Websites are considered to be english websites, if percentage of english words > 50 %

Most frequent words list for category

Most frequent words list for each category is necessary to determine key words that are commonly used in the particular category.

Process of most frequent words list for each category creation:

1. Lists with all words of websites for each category are created.
2. Words for each category are counted and sorted in ascending mode.
3. Filter 2500 most frequent words in each category list.

Machine Learning

1. Features and Labels creation
2. Custom Model
3. Machine Learning Models

Features and Labels creation

- **Features** are binary lists of all websites words in each category which occur in *Most frequent words list* for category.
- **Labels** are lists of websites categories value

Example of Feature and Labels generation

Website: 'https://www.vu.lt/en/'
Website category: 'Career_and_Education'
Most frequent words list: ['student', 'school', 'program', 'university', 'resource', 'service',...]
Website most frequent words: ['university', 'vilnius', 'study', 'international', 'faculty', 'student',...]
Feature list: [1, 0, 0, 0, 0, 1,...]
Label value: [5]

Custom model

Custom model determines website category by estimating features weights.

- Features are calculated for all categories.
- Weight of each feature is calculated by formula:

  weight += 2500 - index where feature value equals 1

- Category with the highest feature weight sum would be applied to the website

Machine Learning Models

Machine learning models:

- Logistic Regression
- Linear Support Vector Classification

- Machine Learning models are created using **sklearn** library.
- Training features and labels lists are fit into models.
- Testing features and labels samples was used for models testing.

Results (1)

Performance results:

1. Open source Data Set:
   - **Logistic Regression model**:
     - Accuracy score: 0.85
     - Precision score: 0.85
     - Recall score: 0.79
     - F1 score: 0.81
   - **Linear Support Vector model**:
     - Accuracy score: 0.79
     - Precision score: 0.74
     - Recall score: 0.72
     - F1 score: 0.73
2. Custom Data Set:
   - **Custom model**: 69.5 % (197 correct predictions of 282)
   - **Logistic regression model**: 57.8 % (163 correct predictions of 282)
   - **Linear support vector model**: 52.5 % (148 correct predictions of 282)

Results (2)

Logistic Regression model results for each category:

	Logistic Regression		
Category	**precision**	**recall**	**f1-score**
Recreation_and_Hobbies	0.98	0.86	0.91
Food_and_Drink	0.84	0.86	0.85
News_and_Media	0.86	0.92	0.89
Travel	0.87	0.66	0.75
Career_and_Education	0.93	0.80	0.86
Home_and_Garden	0.88	0.86	0.87
Internet_and_Telecom	0.87	0.81	0.84
Gambling	0.80	0.90	0.85
Books_and_Literature	0.87	0.95	0.91
Science	0.95	0.88	0.91
Health	0.82	0.59	0.69
Autos_and_Vehicles	0.94	0.89	0.92
Finance	0.80	0.86	0.83
Sports	0.87	0.85	0.86
Adult	0.81	0.83	0.82
Pets_and_Animals	0.92	0.90	0.91
Business_and_Industry	0.80	0.90	0.85
People_and_Society	0.83	0.26	0.40
Law_and_Government	0.89	0.87	0.88
Beauty_and_Fitness	0.94	0.91	0.93
Arts_and_Entertainment	0.76	0.79	0.78
Games	0.88	0.81	0.84
Reference	0.85	0.76	0.80
Computer_and_Electronics	0.25	0.11	0.15
Shopping	0.94	0.85	0.89

Results (3)

Linear Support Vector Machine model results for each category:

	Linear Support Vector Machine		
Category	**precision**	**recall**	**f1-score**
Recreation_and_Hobbies	0.78	0.80	0.79
Food_and_Drink	0.81	0.78	0.79
News_and_Media	0.81	0.87	0.84
Travel	0.79	0.54	0.64
Career_and_Education	0.83	0.72	0.77
Home_and_Garden	0.79	0.84	0.8
Internet_and_Telecom	0.76	0.72	0.74
Gambling	0.76	0.83	0.79
Books_and_Literature	0.86	0.88	0.87
Science	0.86	0.86	0.86
Health	0.62	0.52	0.56
Autos_and_Vehicles	0.86	0.84	0.85
Finance	0.76	0.78	0.77
Sports	0.73	0.75	0.74
Adult	0.77	0.75	0.76
Pets_and_Animals	0.88	0.88	0.88
Business_and_Industry	0.79	0.85	0.82
People_and_Society	0.36	0.21	0.27
Law_and_Government	0.82	0.82	0.82
Beauty_and_Fitness	0.86	0.78	0.82
Arts_and_Entertainment	0.72	0.74	0.73
Games	0.71	0.73	0.72
Reference	0.75	0.65	0.70
Computer_and_Electronics	0.08	0.11	0.09
Shopping	0.81	0.77	0.79

Questions

Performance scores (1)

- **True Positives (TP)** - is an outcome where the model correctly predicts the positive class. Example of true positive condition: *Person with disease was diagnosed a disease*.
- **True Negatives (TN)** - is a true negative is an outcome where the model correctly predicts the negative class. Example of true negative condition: *Person with no disease was not diagnosed a disease*.
- **False Positives (FP)** - is a result that indicates a given condition exists, when it does not. Example of false positive condition:: *Healthy person was diagnosed with a specific disease*.
- **False Negatives (FN)** - is a test result that indicates that a condition does not hold, while in fact it does. Example of false negative condition:: Person with disease was diagnosed with no disease.

Performance scores (2)

Accuracy score

Classification accuracy is the number of correct predictions made as a ratio of all predictions made.

The accuracy score is calculated by formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Recall score

Recall score is the number of true positives divided by the number of true positives plus the number of false negatives.

The recall score is calculated by formula:

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

Performance scores (3)

Precision score

Precision evaluation method determines of how precise/accurate machine learning model of how many positives predictions have been predicted of total predictions. Precision score is calculated by formula:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

Precision is a good measure to determine, when the costs of False Positive is high.

F1 score

F1 score is the harmonic mean of precision and recall taking both metrics into account. F1 score is calculated by formula:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

LR vs LSVM

- LSVM try to maximize the margin between the closest support vectors while LR the posterior class probability. Thus, LSVM find a solution which is as fare as possible for the two categories while LR has not this property.
- LR is more sensitive to outliers than LSVM because the cost function of LR diverges faster than those of LSVM
- Logistic Regression produces probabilistic values while SVM produces 1 or 0. So in a few words LR makes not absolute prediction and it does not assume data is enough to give a final decision.

SVM vs LSVM

LSVM uses different parameter kernel='linear', so it has more flexibility in the choice of penalties and loss functions and should scale better to large numbers of samples.

Further improvements

1. Download content of website recursively.
2. Revise website categories in the original data set.
3. Translate non english content websites to english language.
4. Improve custom data set.
5. Revise categories.

Results with TOP x most frequent words list

TOP 500 most frequent words

- LR : 0.821501014198783
- LSVM: 0.7505070993914807

TOP 2000 most frequent words

- LR : 0.8643871341640105
- LSVM: 0.7913648217907853

TOP 5000 most frequent words

- LR : 0.8591712547087801
- LSVM: 0.792813677195015 9

TOP 10000 most frequent words

- LR : 0.8487394957983193
- LSVM: 0.7847000869313242