MAY 2020

IDS 566

VIMAL PRAKASH
LAILA HUSSEIN
AJAY SRIVATS
KAVYA RAVI GOWDA

02

ABOUT

The objective of this project is to train different classifiers and test the outcomes to acheive a high accuracy.

WHAT WE'VE DONE

During the course of this project, we've tried four main models : Logistic Regression, SVM, Naive Bayes & Deep Learning.
The next several pages dive deep to disclose the strategies implemented and the results we were able to derive

03

STEP 1: IMPORTING DATA

These were the steps taken to ensure that the data is in a usable format:

1. Data is imported as data_train and data_test
2. The imported data was then converted into a data frame under train_text and test_text
3. We renamed columns under text and category for train_text, test_text, train_target and test_target

STEP 2: DATA PRE-PROCESSING

1. All letters were converted to lower case
2. Punctuation, accent marks and other diacritics were removed
3. Redundant articles were eliminated
4. Stop words removed
5. Lemmatized words
6. Tested accuracy using Unigram and Bigram
7. Redundant articles

04

STEP 3: CHALLENGES ENCOUNTERED

1. The dataset is quite messy and there are a lot of redundant words and characters which were to be removed
2. The dataset is quite large which takes longer time in training the models
3. Finding best hyperparameters was a challenge due to limited computing power
4. Small memory in the computer making slower to run the code.

STEP 4: HOW WERE THEY RESOLVED?

1. The first problem as tackled by removing the redundant articles in the pre-processing step.
2. Since it is a large data set, we did as much pre-processing as possible so the training models would take less time to run by following step 2.
3. We found the hyper-parameters by doing a grid search.
4. Using a computer with a bigger memory in the future.

STEP 5: THE MODELS

01 Multinominal Naive Bayes

The multi-nominal Naive Bayes classifier is ideal for classification for discrete features.

Accuracy: 80.97%

Precision: 83.00%

accuracy 0.8097450876261285

	precision	recall	f1-score	support
0	0.81	0.66	0.73	319
1	0.78	0.70	0.74	389
2	0.77	0.72	0.75	394
3	0.65	0.79	0.71	392
4	0.85	0.80	0.82	385
5	0.86	0.78	0.82	395
6	0.87	0.78	0.82	390
7	0.88	0.91	0.90	396
8	0.93	0.95	0.94	398
9	0.90	0.93	0.92	397
10	0.89	0.98	0.93	399
11	0.74	0.96	0.84	396
12	0.82	0.64	0.72	393
13	0.92	0.79	0.85	396
14	0.84	0.94	0.89	394
15	0.62	0.96	0.76	398
16	0.65	0.94	0.77	364
17	0.93	0.94	0.93	376
18	0.94	0.49	0.65	310
19	0.95	0.22	0.35	251
avg / total	0.83	0.81	0.80	7532

Implementing multinominal naive results in an accuracy of 80.97% and a precision of 83%. Accuracy using unigram before lemmatization is 80.5% and 80.9% after lemmatization. Using bigram resulted in 80.3% before lemmatization and 80.5% after.

STEP 5: THE MODELS

02 Logistic Regression

Logistic regression is a classification problem used to assign observations to a discrete set of classes

accuracy 0.8177110993096123

	precision	recall	f1-score	support
0	0.83	0.71	0.76	319
1	0.66	0.78	0.72	389
2	0.76	0.77	0.77	394
3	0.69	0.69	0.69	392
4	0.80	0.80	0.80	385
5	0.82	0.76	0.79	395
6	0.75	0.86	0.80	390
7	0.87	0.88	0.88	396
8	0.92	0.95	0.93	398
9	0.86	0.91	0.89	397
10	0.89	0.95	0.92	399
11	0.95	0.91	0.93	396
12	0.73	0.74	0.74	393
13	0.87	0.80	0.84	396
14	0.87	0.90	0.89	394
15	0.78	0.94	0.85	398
16	0.74	0.88	0.81	364
17	0.95	0.87	0.91	376
18	0.87	0.59	0.70	310
19	0.84	0.45	0.59	251
avg / total	0.82	0.82	0.82	7532

Accuracy: 81.77%

Precision: 82.00%

Implementing logistic regression results in an accuracy of 81.77% and a precision of 82.00%. Accuracy using unigram
before lemmatization is 82.6% and 83.0% after lemmatization. Using bigram resulted in 81.7% before lemmatization and 81.9% after.

STEP 5: THE MODELS

03 Stochastic Gradient Modeling

SGM aims to to minimize the empirical risk of a model by repeatedly computing the gradient of a loss function on a single training example, or a batch of few examples, and updating the model parameters accordingly.

accuracy 0.8552841210833776

	precision	recall	f1-score	support
0	0.85	0.77	0.81	319
1	0.76	0.77	0.77	389
2	0.75	0.78	0.76	394
3	0.72	0.73	0.73	392
4	0.83	0.85	0.84	385
5	0.87	0.77	0.82	395
6	0.83	0.92	0.87	390
7	0.94	0.90	0.92	396
8	0.96	0.96	0.96	398
9	0.89	0.94	0.91	397
10	0.93	0.98	0.96	399
11	0.93	0.95	0.94	396
12	0.83	0.78	0.80	393
13	0.90	0.87	0.89	396
14	0.87	0.94	0.90	394
15	0.86	0.95	0.90	398
16	0.75	0.92	0.83	364
17	0.96	0.91	0.94	376
18	0.86	0.64	0.73	310
19	0.80	0.62	0.70	251
avg / total	0.86	0.86	0.85	7532

Accuracy: 85.52%

Precision: 86.00%

Implementing SVM results in an accuracy of 85.52% and a precision of 86.00%. Accuracy using unigram before lemmatization is 84.7% and 84.9% after lemmatization. Using bigram resulted in 85.5% before lemmatization and 85.5% after.

STEP 5: THE MODELS

04 Deep Learning

Deep Learning is a subset of machine learning. Deep artificial neural networks are a set of algorithms.
Implementing Deep Learning resulted in an accuracy of 83.47%.

RESULTS:

Overall, we determined that Stochastic Gradient Boosting works the best and had the highest accuracy of 85.52%

LEARNINGS:

01 What are the advantages of data cleaning in text mining?

Data cleaning is important because it improves the quality of our data. Overall it increases the productivity of our data set. By cleaning out the data, the unnecessary information is removed, keeping us only with high quality data which we can conclude accurate information with. The business value it can have is to minimize costs in the long-term.

09

02 Why is good accuracy important?

Accuracy is a metric for evaluating classification models. A good accuracy tells us that our model is performing well.

03 Does Deep Learning always perform better?

Often it is assumed that deep learning performs the best. During this project, we were able to determine that Deep Learning wasn't the best model.

04 Is a single model ideal for every problem?

A single model isn't ideal for every problem. Different algorithms work better and can generate better accuracies. In this project, we were able to identify that SGD worked best.