

Data Challenge 1: Classification d'articles ArXiv

Team Name on Kaggle : Arjun

Full Name: Arjun Vaithilingam Sudhakar , (20182449-udem student id)

Email id: innovatorarjun@gmail.com/arjun.vaithilingam.sudhakar@umontreal.ca

I. Introduction:

The goal of the challenge is to predict the category of the reaserch paper based on the abstract of the research. We have given a data set for training and validation (7500x2) with which we have to train a generalized model that can predict the category of 15000 articles.

The Approach used are,

1. Visualizing the data (to get better understanding of the data)
2. Data Cleaning (To remove/preprocess the data which makes the data more meaningful)
3. Converting the sentence to Vectors (TFID and Bag of Words)
4. Build models (Naïve Bayes, SVM, logistic Regression)
5. Fine tuning the hyper parameters using grid search

With Bernoulli Naïve Bayes, in Kaggle we got 78.86% and able to beat the **Random & NB Baseline**.

[submission_V3 \(1\).csv](#)

15 days ago by Arjun

0.78542

0.78822



Bernoulli Naive Bayes implementation from the scratch

With Multinomial Naïve Bayes with TFIDF, could able to achieve and able to beat the **TA Best accuracy**,

Public Leaderboard:

16	Arjun		0.81511	21	12d
----	-------	--	---------	----	-----

Private Leaderboard:

33	▼ 17	Arjun		0.80428	21	12d
----	------	-------	--	---------	----	-----

II Feature Design:

Text data generally contains lot of noise (punctuation's, stop words) which might mislead the model. Hence it is essential to do pre-processing before feeding the data to the model.

The approaches used in pre-processing are,

a. Lower Case:

- Let's start with an Example, "Energy" and "energy" are considered as the different by the system. However, in real word they mean the same. Make it to lower case, the system will recognize both words as "energy".
- The accuracy of the model would be badly affected if we don't make the text to lower case and make it meaningful so that the model learns valuable information and won't be mis leaded.

b. Punctuations:

- In real word, we generally use punctuation to make it easy and add meaning to the readers. However, for machine learning models, punctuations are considered as a noise in terms of text data and it does not contribute any value/meaning to the model.

c. Dropping length(word)==1:

- After removing punctuations and making it lower, I observe the word of length 1 doesn't add any value for the predictions. Eg: (a-z characters).
- After dropping the word of length one, the number of words in the vocabulary got reduced tremendously and increased the accuracy of the model.

d. Stop Words:

- Eg of stop words → and, that, they, what, a, the
- In English these words are used often with high occurrences in a sentence. The model should not give weightage to these words and it can't predict correctly as it confuses the model which might cause misclassification.

Before Cleaning:

Id		Abstract	Category
0	0	The energy released in a solar flare is partitioned between thermal and non-thermal particle energy and lost to thermal conduction and radiation over a broad range of wavelengths. It is difficult to determine the conductive losses and the energy radiated at transition region temperatures during the impulsive phases of flares. We use UVCS measurements of O VI photons produced by 5 flares and subsequently scattered by O VI ions in the corona to determine the $5.0 < \log T < 6.0$ transition region luminosities. We compare them with the rates of increase of thermal energy and the conductive losses deduced from RHESSI and GOES X-ray data using areas from RHESSI images to estimate the loop volumes, cross-sectional areas and scale lengths. The transition region luminosities during the impulsive phase exceed the X-ray luminosities for the first few minutes, but they are smaller than the rates of increase of thermal energy unless the filling factor of the X-ray emitting gas is > 0.1 . The estimated conductive losses from the hot gas are too large to be balanced by radiative losses or heating of evaporated plasma, and we conclude that the area of the flare magnetic flux tubes is much smaller than the effective area measured by RHESSI during this phase of the flares. For the 2002 July 23 flare, the energy deposited by non-thermal particles exceeds the X-ray and UV energy losses and the rate of increase of the thermal energy.	astro-ph
1	1	In light of current atmospheric neutrino oscillation data, we revisit the invisible decay of the standard model Higgs boson and other pseudoscalar mesons which can be enhanced because of large number of KK modes in models with right-handed singlet neutrinos in large extra dimensions. We find that the invisible decay rate of Higgs can be as large as $H \rightarrow h\bar{b}b$ decay rate only for a very restricted region of parameter space. This parameter space is even further restricted if one demands that the dimensionless neutrino Yukawa coupling λ is $O(1)$. We have also studied the scenarios where singlet neutrino propagate in a sub-space, which lowers the string scale M_* and keeps neutrino Yukawa coupling $O(1)$. We have also considered decays of other spin-0 mesons to $\nu\bar{\nu}$ and found the rates to be too small for measurement.	hep-ph

After Cleaning:

Id		Abstract	Category
0	0	energy released solar flare partitioned thermal non thermal particle energy lost thermal conduction radiation broad range wavelengths difficult determine conductive losses energy radiated transition region temperatures impulsive phases flares use uvcs measurements vi photons produced flares subsequently scattered vi ions corona determine log transition region luminosities compare rates increase thermal energy conductive losses deduced rhesi goes ray data using areas rhesi images estimate loop volumes cross sectional areas scale lengths transition region luminosities impulsive phase exceed ray luminosities first minutes smaller rates increase thermal energy unless filling factor ray emitting gas estimated conductive losses hot gas large balanced radiative losses heating evaporated plasma conclude area flare magnetic flux tubes much smaller effective area measured rhesi phase flares july flare energy deposited non thermal particles exceeds ray uv energy losses rate increase thermal energy	astro-ph
1	1	light current atmospheric neutrino oscillation data revisit invisible decay standard model higgs boson pseudoscalar mesons enhanced large number kk modes models right handed singlet neutrinos large extra dimensions find invisible decay rate higgs large bar decay rate restricted region parameter space parameter space even restricted one demands dimensionless neutrino yukawa coupling also studied scenarios singlet neutrino propagate sub space lowers string scale ast keeps neutrino yukawa coupling also considered decays spin mesons nu bar nu found rates small measurement	hep-ph

e. TFIDF/Bag of Words:

- We convert the text data into number understandable by the model, hence we use TFIDF and Bag of Words methodologies to convert the text data into meaningful number vectors. We used a dictionary to hold the occurrences and sorted in descending order to get top vocabulary.
- Label Encoding is done to the Category (prediction/ dependent variable)

III. Algorithms used:

The algorithms that were used for this problem were

1. Multinomial Naïve Bayes
2. Support Vector Classifier
3. K Nearest Neighbour
4. Random Forest

a. Multinomial Naïve Bayes:

- This algorithm basically captures the term frequency i.e. the number of occurrences of a word in the whole document. Then it will normalize after which the maximum likelihood estimation is done based on the training data to estimate the conditional probability.
- Its is based on the probabilistic approach
- **Highlights:** this algorithm runs quick and good even the dataset size is low and less hyper parameter to tune.

b. Support Vector Classifier:

- Support Vector Classifier tries to find the hyper plane that separates the class in an N-dimensional space. EG: some classes are not separable in 2D, however if we move to 4 or 5 D then we can separate the classes using the hyperplane.
- SVC are not influenced by the outliers in the data and majorly depends on the support vector points near the hyper plane. The higher the margin then higher the confidence in the classification
- It is a deterministic approach and one of the powerful linear and non-linear algorithms
- **Highlights:** This algorithm gave good results, however some time it overfits due to higher complexity that the model has and runs pretty quick

c. K Nearest Neighbour:

- KNN mainly depends on the distance metrics (like Euclidean/- Manhattan's). Hence, I personally think, it might suffer from outliers and need as good amount of pre-processing
- We could reduce the dimensions through PCA or LDA or any other dimensionality approach, we could somewhat control the dimensionality curse in KNN algorithms for better performance.
- There is no as such training time for the KNN algorithms and the model doesn't learn any parameters and mainly depends of the hyper parameters.
- **Highlights:** Simple and straight forward algorithm, however finding the correct k value is quite complex and a small change in k value can change the class of prediction.

d. Random Forest Algorithms:

- The intuition is the collection of decision tree and using the concept of averaging technique or crowd of wisdom.
- Each individual tree constitutes to some output and based on the majority vote the class of the output is determined which protects each tree from their individual errors.
- **Highlights:** It took a long time to run the Radom Forest Algorithms when compared with the other models due to the large features used for prediction. Finding the hyper parameters were difficult and random forest were not preforming good for this problem and there was a higher chance of overfitting most of the time

IV Methodology:

a. Train/validation split:

- It is essential to split the training dataset into train and validation to choose the right hyper parameters. Once the model is built using the right hyper parameters use can it to predict the test set. The main advantage is that, it removes the bias towards the selection of hyper parameters
- The split should be random, so that we can get a good estimation of the empirical risk

b. Grid Search:

- It is an amazing approach to choose the right hyper parameters for the model.
- We can create a dictionary and add the value of the hyper parameters. In the back, it generally acts like a for loop and fit the model with all the combination of the hyper parameters and gives us the best_params.
- Finding the right parameters plays an important role in improving the accuracy and indeed it is true. In our problem, we saw a good increase in the accuracy when we use the right hyperparameters.

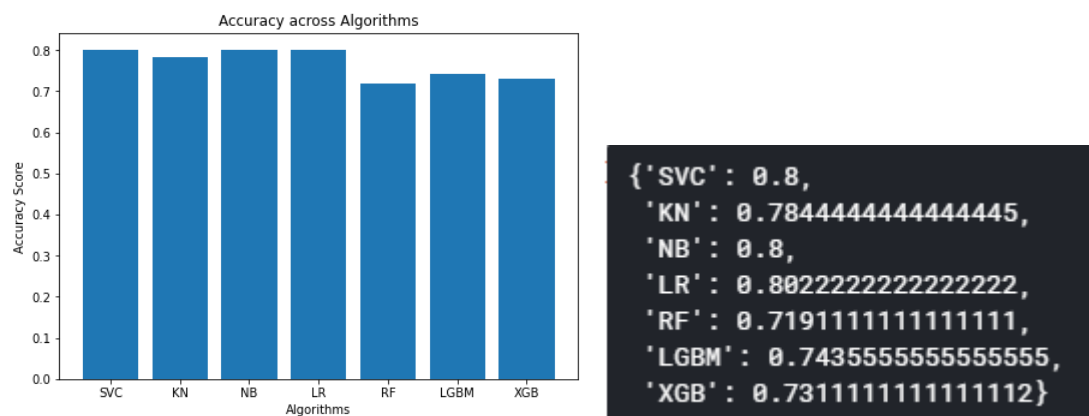
c. Cross validation:

- In cross validation, we generally reserve a portion of sample data set on which we won't perform the training and we do the test on the reserved dataset.
- We can do cross validation multiple time and take an average to get some what precise accuracy and we can know about the standard deviation on the accuracy

V Results:

SVM and Naïve Bayes algorithm was performing good as per the validation set. Random forest algorithms and its variants weren't giving promising results. When SVM submission.csv file is submitted, the accuracy on the test dataset(public) was poor as the model was overfitting with the training data even after tuning, the results didn't surpass Multinomial Naïve Bayes with TFIDF,

Accuracy Report in validation data set



Accuracy in Test Set (After uploading in Kaggle):

With Bernoulli Naïve Bayes, in Kaggle we got 78.86% and able to beat the **NB Baseline**.

[submission_V3 \(1\).csv](#)

15 days ago by Arjun

Bernoulli Naive Bayes implementation from the scratch


0.78542

0.78822




With Multinomial Naïve Bayes, could able to achieve and able to beat the **TA Best accuracy**,

Public Leaderboard:

16	Arjun		0.81511	21	12d
----	-------	---	---------	----	-----

Private Leaderboard:

33	▼17	Arjun		0.80428	21	12d
----	-----	-------	---	---------	----	-----

SNo	Algorithm	Hyper parameter
1.	Bernoulli Naïve Bayes	Alpha
2.	Multinomial Naïve Bayes	Alpha
3.	SVC	C, kernel
4.	Random Forest	Max_depth, gamma, n_estimator

VI Discussion:

a. Pros:

- Pre-processing at the start helps to reduce the vocabulary size
- Train/val/test approach helps to select the hyper parameters without bias
- Experimented many algorithm's which helped to short list the algorithm for final prediction
- Grid search played a key role in finding the right hyper parameters

b. Cons:

- Performed even better in Public test dataset, however there is a huge drop in the private leader board, the model should be even generalized.
- It was a time-consuming process to build the final model
- It was based on trail and error approach and there is no guarantee that one algorithm is always the best.

c. Idea for improvements:

- Have to get the domain expertise and identify important features which gives a better result and reduce huge time in unnecessary area.

VII Statement of contributions:

I hereby state that all the work presented in this report is that of the author

VIII References:

1. <https://machinelearningmastery.com/gentle-introduction-bag-words-model/>
2. <https://blog.datasciencedojo.com/unfolding-naive-bayes-from-scratch-part-1/>
3. <https://www.kaggle.com/abhishek/approaching-almost-any-nlp-problem-on-kaggle>
4. https://en.wikipedia.org/wiki/Naive_Bayes_classifier
5. <https://towardsdatascience.com/complete-guide-to-pythons-cross-validation-with-examples-a9676b5cac12>
6. <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>
7. <https://builtin.com/data-science/random-forest-algorithm>
8. <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>