समाचार सुर्खियों के वर्गीकरण के लिए एक मशीन लर्निंग (A Machine Learning Approach for News Headlines Classification)

जसमीत सिंह[1], अनुभव अग्रवाल[2], डॉ. कपिल[3]
(Jasmeet Singh[1], Anubhav Aggarwal[2], Dr. Kapil[3])

NIT Kurukshetra, Kurukshetra 136119, India
[1]jasmeet45.js@gmail.com, [2]anubhavagg93@gmail.com, [3]kapil@nitkkr.ac.in

सार

इस लेख में हम समाचार सुर्खियों के लिए एक वर्गीकरण सॉफ्टवेर का निर्माण पर विचार करेंगे. हमने तीन अल्गोरिथ्म्स का प्रयोग किया है इसे बनाने के लिए . हालाँकि सारे अल्गोरिथ्म्स ने हमें ९०% से ज्यादा एक्यूरेसी दी है, हमें सबसे बेहतर परिणाम सपोर्ट वेक्टर मशीन से मिले हैं.

Keywords: svd; naïve bayes; svm; logistic regression;

12. परिचय

Technology के आने से हमें काफी सारा data मिला है जो डिजिटल रूप में है . ये data अनेक प्रकार से हमारे सामने आता है जैसे आवाज़, चित्र , लेख या फिर internet पे वित्रत किसी भी प्रकार के लेख. हालाँकि ये सब काफी बड़ी मात्रा में उत्पन्न हो रहे हैं लेकिन इनमे लेख(text data) काफी तेजी से उत्पन्न होता है.

इन लेखो में किसी भी प्रकार का वर्गीकरण मौजूद नही होता है इसीलिए इन्हें संभालना काफी मुश्किल होता है . इन्हें अलग अलग वर्गो में बांटना जैसे की खेल, शिक्षा, राजनीति इत्यादी में बाँटना काफी मुश्किल हो जाता है. अभी तक इन्हें मानव श्रम के साथ ही बांटा जा सकता है . हम इन्हें मशीन लर्निंग का प्रयोग करके कंप्यूटर द्वारा बांटना चाहते हैं. मशीन लर्निंग लगाने के बाद इनपर labelling और tagging करना काफी आसन और सस्ता हो जायेगा.

मशीन लर्निंग उस कार्य को कहते हैं जिनमे हम कंप्यूटर को कोई काम करना सिखाते हैं बिना उसने कोई मुखर निर्देश दिए. El Naqa [1] ने मशीन लर्निंग को कुछ ऐसे परिभाषित किया है "Machine learning is an evolving branch of computational algorithms that are designed to emulate human intelligence by learning from the surrounding environment". मशीन लर्निंग को हम मुख्यतः दो भागो में बाँट सकते हैं : supervised और unsupervised. supervised लर्निंग में ट्रेनिंग के दौरान हम कंप्यूटर को उचित लेबल्स देते हैं जिनसे वे सीश्ते हैं. unsupervised लर्निंग में कंप्यूटर अपने आप ही dataset से लेबल्स निकलते हैं और उनसे सीखते हैं. इसमें कंप्यूटर dataset में मौजूद सारे छिपे पैटर्न्स को ढूंढ़ते हैं .

हमने लेख वर्गीकरण में कई लेख वर्गीकरण अल्गोरिथ्म्स का प्रयोग किया है. हमें पता चला है की इनमे से कई तकनीक का इस्तेमाल हम लेख वर्गीकरण के लिए कर सकते हैं. . R. Jindal [3] ने लेख वर्गीकरण को कुश इस प्रकार परिभाषित किया है

"Text categorization is a task of assigning one or more predefined categories to the analysed document, based on its content."

13. प्रेरणा

इस कार्य को करने का प्रमुख उद्देश्य था की हम मानव श्रम को काम से काम प्रयोग में लायें labelling और tagging के लिए. ये अनुमान लगाया गया है की textual data सन 2020 तक ककरीब 20 zetabytes तक पहुँच जायेगा. इस ऑटोमेशन की वाजह से करोडो की संख्या में प्रकाशित हो रहे लेखो को लेबल किया जा सकता है वो भी अपने आप . अगर हम इस सॉफ्टवेर को असली समाचार लेखो से ट्रेन करें तो हम असली समाचार लेखो को उसी समय वर्गीकृत कर सकते हैं .

इसका एक और प्रयोग किया जा सकता है की हम किसी समाचार लेख को असली या नकली बता सकते हैं. इससे कानूनी संस्थानों को ये पता लगाने में मदद मिलेगी की कोई कहानी स्ली है या नकली ।

14. अल्गोरिथम

हमारा अल्गोरिथम supervised लर्निंग पर आधारित है . इसके लिए हमें निम्न विधि के आधार पर कार्य करना होगा :

14.1. Dataset Collection

इस चरण में हमने वर्गीकरण के प्रशिक्षण के लिए डेटासेट एकत्र करने पर ध्यान केंद्रित किया. हमारे कार्य के लिए हमने कैलिफोर्निया विश्वविद्यालय इरविन से समाचार पत्रों का एक डेटासेट एकत्र किया।

14.2. Stop words removal

इस चरण में, हमने सबसे अधिक उपयोग किए जाने वाले शब्दों को हटा दिया जैसे as a, an, with, the, इत्यादि।

14.3. Stemming

हमें दस्तावेजों में शब्दों की जड़ या तना मिला। उदाहरण के लिए पुराने से पुराने को बदलना। (older to old)

14.4. Bag of World representation

इस चरण में, हमने दिए गए दस्तावेजों को प्रत्येक अलग-अलग शब्द के लिए शब्द-आवृत्ति की जोड़ी के रूप में दर्शाया।

14.5. Train-Test Split

इस चरण में, हमने प्रशिक्षण के लिए 80% डेटा और मूल्यांकन के लिए 20% असाइन किया।

14.6. Training of the classifier

इस चरण में, हमने मॉडल के निर्माण के लिए मशीन लर्निंग एल्गोरिदम को डेटा खिलाया जिसका उपयोग न्यूज़ हेडलाइंस के टैग या लेबल की भविष्यवाणी करने के लिए किया जाएगा।

14.7. Evaluation of the classifier

इस चरण में, हमने अपने कार्य के लिए विभिन्न मशीन लर्निंग तकनीकों की सटीकता को माप लिया ।

Text categorization के इस अल्गोरिथम से हमने जो मॉडल का निर्माण किया है उसमे SVM का प्रयोग किया गया है |
" Support Vector Machine" एक supervised मशीन लर्निंग अल्गोरिथम है जिसका उपयोग Classification और Regression दोनों कार्यो के लिए किया जा सकता है | हालाँकि, इसका प्रायः प्रयोग Classification में ही किया जाता है | इस अग्लोरिथ्म में हम किसी भी data item को एक n-dimensional space में plot किया जाता है (जहा पे n हमारे features की संख्या दर्शाती है) जहाँ हर एक हर एक feature का मान किसी coordinate के मान के बराबर होती है |
Support vector machine का classification कुछ इस प्रकार कार्य करता है : पहले हम data को इस प्रकार बांटते है की दो या अधिक features classified मॉडल में intersect करते हैं | ये intersection कुछ इस प्रकार से किया जाता है की ये सबसे best suited label निर्धारित किया जा सके | svm ट्रेनिंग data को , जिन्हें support vectors कहते हैं , उन्हें दो या दो से अधिक भागो में बांटती है | ये बंटवारा classification method और feature selection तकनीक पर निर्भर होता है साथ में ये इस बात पर भी निर्भर होती है की labels की संख्या कितनी है | इन support vectors से अलग अलग प्रकार से

Hyperplanes बनते है जिन्हें margins कहा जाता है | margins हमें hyperplanes को support vectors के बीच में डालने में मदद करते है साथ ही साथ data points को और सटीकता से साथ classify करने में मदद करते है | जो support vectors हमें मिलते हैं इन्हें रक multi – dimensional graph में plot किया जाता है | इस graph का हरेक dimension एक feature को दर्शाता है | इसके बाद इन्हें अलग अलग sets में बाँट दिया जाता है सुर इन sets का प्रयोग किसी भी document का label निर्धारित करने में किया जाता है | ये बंटवारा करने के लिए support vectors के बीच में कई hyperplanes डाले जाते हैं | ये hyperplanes पहले से ही svm के द्वारा किसी feature selection technique का प्रयोग करके निकले जा चुके होते हैं | ये hyperplanes सारे support vectors से बराबर दूरी पर होते है | एक hyperplane एक plane का सामान्यीकरण है।

Data को बांटने के लिए hyperplane को लगाने के कई तरीके हैं जिनसे हमें अच्छी सटीकता मिले :
पहला तरीका : निम्न दर्शाए चित्र में 3 hyperplanes को एक ही dataset में दिखाया गया है | hyperplane डालने का एकमात्र नियम ये है की hyperplane दो या दो से अधिक features का बंटवारा करे | नीचे दर्शाए चित्र में वो hyperplane 'B' है |

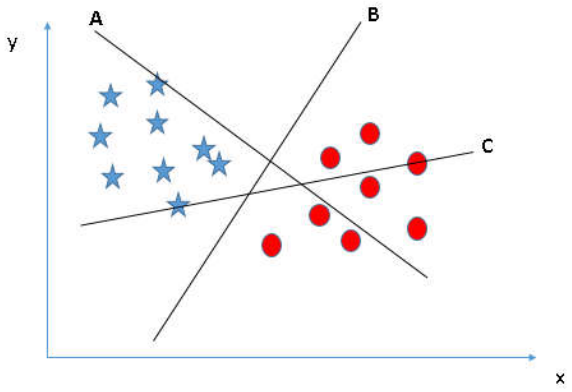

Fig. 1. hyperplane 1

दूसरा तरीका : इसमें कई सारे hyperplanes एक ही dataset पे लगे होते है और सब बराबर सटीकता दे रहे होते हैं | ऐसे में हम उस hyperplane को चुनते है जो दोनों vector sets से बराबर दूरी पर होता है | नीचे दर्शाए चित्र में ये hyperplane 'C' है |

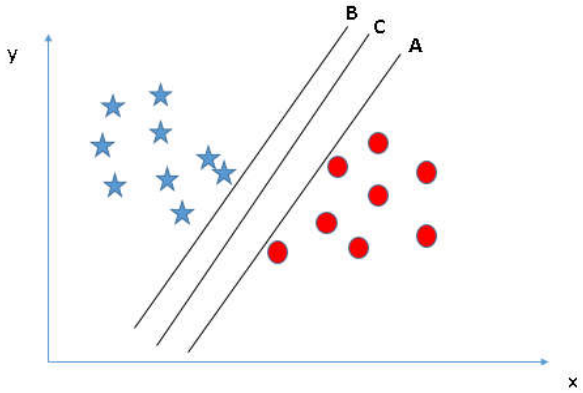

Fig. 2. hyperplane 2

तीसरा तरीका : इसमें hyperplane की margins तो ज्यादा होते हैं पर ये hyperplane dataset को अच्छी सटीकता से

वर्गीकृत नही कर पाटा है | इस अवस्था में हम दुसरे hyperplanes को भी देखते है ताकि उनकी भी सटीकता का आकलन किया जा सके और सबसे अच्छे hyperplane को निर्धारण किया जा सके |

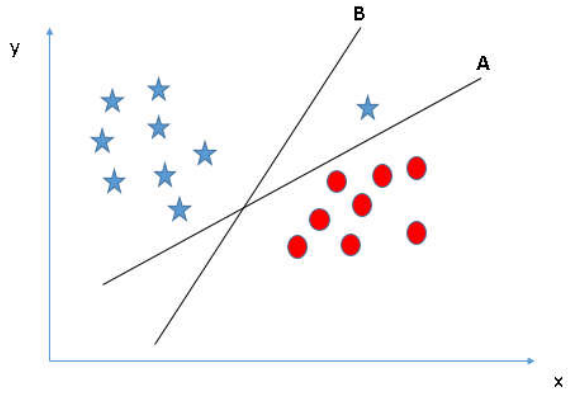

Fig. 3. hyperplane 3

चौथा तरीका : इसमें एक vector गलत जगह पे आधारित होता है | इसकी वजह से hyperplane को डालने में मुश्किल होती है | इस अवस्था में उस गलत vector को एक glitch का गलती की तरह देखा जाता है और इसे संपूर्ण रूप से अनदेखा किया जाता है | ऐसे vectors को 'outliners' कहते हैं |

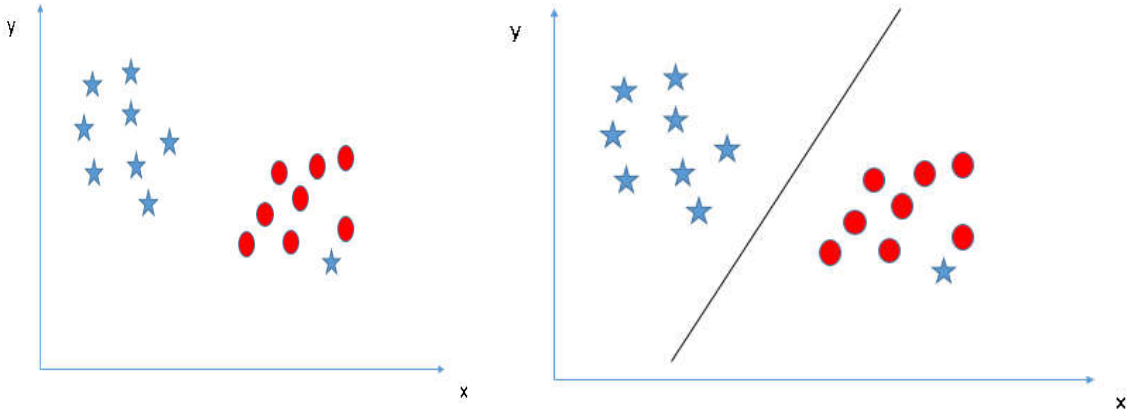

Fig. 4. (a) hyperplane 4; (b) hyperplane 5

graph के अन्दर optimal hyperplanes की उत्पत्ति इसे बनाने में लिए गए कदम और इस्तेमाल की गयी technique का प्रयोग किया जाता है | इसका तरीका नीचे लिखा हुआ है :

ये वो equation है जिसका प्रयोग 2 भागो में बंटे features के लिए hyperplane बनाने के लिए किया जाता है |

$$f(x) = \beta_0 + \beta^T x \quad \ldots. (1)$$

जहा : $\beta = weight\ bias, \beta_0 = bias$

एक graph अनंत hyperplanes बना सकता है और इनका प्रयोग data को वर्गीकृत करने के लिए किया जा सकता है लेकिन सबसे सटीक hyperplane को ये निम्न शर्ते माननी पड़ेगी :

$$|\beta_0 + \beta^T x| = 1 \quad \ldots. (2)$$

जहाँ x उन vector points को दर्शाता है जो hyperplane के सस्बे नज़दीक हैं | इस representation को canonical representation कहते हैं |

$$Distance = \frac{|\beta_0+\beta^T x|}{\|\beta\|} \quad \ldots. (3)$$

eqn 2 का आगे प्रयोग margins निकलने में किया जाता है |
equation 2 और 3 का प्रायोग करके हम निम्न आकलन पर आते हैं :

$$Distance_{Support\ vectors} = \frac{|\beta_0+\beta^T x|}{\|\beta\|} = \frac{1}{\|\beta\|} \quad \ldots. (4)$$

चूँकि margin सबसे कम और बराबर दूरी होती है नजदीकी support vectors और hyperplanes के बीच में , इसीलिए अलग अलग features के vectors के बीच में दूरी इसके बराबर होती है :

$$D = \frac{2}{\|\beta\|} \quad \ldots. (5)$$

अंततः D को maximize करने के लिए हम $L(\beta)$ को मिनीमाइज करते हैं जो कुछ शर्तों (Constraints) पे निर्धारित होता है | ये कंस्ट्रेंट्स hyperplane के requirements को मॉडल करते हैं जिससे ट्रेनिंग examples को अच्छे से क्लास्सिफ्य किया जा सकता है :

$$\min_{\beta,\beta_0} \quad L(\beta) = \frac{1}{2}\|\beta\|^2 \ subject\ to\ y_i(\beta^T x_i + \beta_0) \geq 1\ \forall i \quad \ldots. (6)$$

यहाँ y_i ट्रेनिंग example को दर्शाती है |
Langrangian optimization की दिक्कत langrange multiplier का प्रयोग करके हासिल हुए weight vector β और bias β_0 का प्रयोग करके निवारण किया जा सकता है

15. तुलनात्मक विश्लेषण

हमने तीन machine learning approaches का प्रयोग किया है, सबमे एक बार SVD का रयोग किया गया है और एक बार नही| इनके परिणाम निम्नलिखित हैं :

Algorithm	With SVD [8]	Without SVD
Multinomial Naive Bayes [5]	90%	92%
SVM [6]	94%	95%
Logistic Regression [7]	92%	94%

16. निष्कर्ष और भविष्य के दायरे

लगभग 4 लाख समाचार पत्रों के बड़े dataset का प्रयोग करके प्रशिक्षण के बाद, सटीकता 100% के करीब पहुंच जाती है। 100% सटीकता तक पहुंचना व्यावहारिक रूप से संभव नहीं है, हम अब तक deep learning का उपयोग करके हासिल किए गए कार्यों से अधिक सटीकता तक पहुंच सकते हैं। लेकिन deep learning सबसे बड़ी गड़बड़ी है कि प्रशिक्षण के लिए इसे एक विशाल डेटासेट की आवश्यकता है। लेबल किए गए डेटासेट को इकट्ठा करना बहुत बड़ा सिरदर्द है और हम प्रशिक्षण के लिए अलग-अलग या लेबल किए गए डेटासेट की थोड़ी मात्रा के साथ बिना लेबल किए गए डेटासेट को लागू करने के लिए और अधिक शोध कर सकते हैं।

References

1. El Naqa I., Murphy M.J. (2015) What Is Machine Learning?. In: El Naqa I., Li R., Murphy M. (eds) Machine Learning in Radiation Oncology. Springer, Cham
2. Anubhav A., Jasmeet S., Dr. Kapil. (2018) A Review of Different Text Categorization Techniques. International Journal of Engineering & Technology, 7 (3.8) (2018) 11-15
3. R. Jindal, R. Malhotra, A. Jain (2015), "Techniques for text classification: Literature review and current trends", Webology, Volume 12, Number 2
4. John Gantz and David Reinsel (2012) THE DIGITAL UNIVERSE IN 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East. Technical Report 1. IDC, 5 Speen Street, Framingham, MA 01701 USA
5. Kibriya A.M., Frank E., Pfahringer B., Holmes G. (2004) Multinomial Naive Bayes for Text Categorization Revisited. In: Webb G.I., Yu X. (eds) AI 2004: Advances in Artificial Intelligence. AI 2004. Lecture Notes in Computer Science, vol 3339. Springer, Berlin, Heidelberg
6. Joachims T. (1998) Text categorization with Support Vector Machines: Learning with many relevant features. In: Nédellec C., Rouveirol C. (eds) Machine Learning: ECML-98. ECML 1998. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), vol 1398. Springer, Berlin, Heidelberg
7. Alexander Genkin, David D Lewis & David Madigan (2007) Large-Scale Bayesian Logistic Regression for Text Categorization, Technometrics, 49:3, 291-304, DOI: 10.1198/004017007000000245
8. Golub, G.H. & Reinsch, C. Numer. Math. (1970) 14: 403. https://doi.org/10.1007/BF02163027