Über Google in analyticsvidhya.com anmelden   ✕

Daniel Vogel
dididerdenker@googlemail.com

Als Daniel fortfahren

Zum Erstellen Ihres Kontos gibt Google Ihren Namen, Ihre E-Mail-Adresse und Ihr Profilbild an analyticsvidhya.com weiter. Weitere Informationen finden Sie in der Datenschutzerklärung und den Nutzungsbedingungen für analyticsvidhya.com.

NEXT=HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/2020/02/QUICK-INTRODUCTION-BAG-OF-WORDS-BOW-TF-IDF/)

☰

Analytics Vidhya (https://www.analyticsvidhya.com/blog/)

BEGINNER (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/BEGINNER/)

NLP (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/NLP/)

TECHNIQUE (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/TECHNIQUE/)

TEXT (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/TEXT/)

UNSTRUCTURED DATA (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/UNSTRUCTURED-DATA/)

Quick Introduction to Bag-of-Words (BoW) and TF-IDF for Creating Features from Text

PURVA HUILGOL (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/AUTHOR/PURVA91/), FEBRUARY 28, 2020 LOGIN TO BOOKMARK THIS …
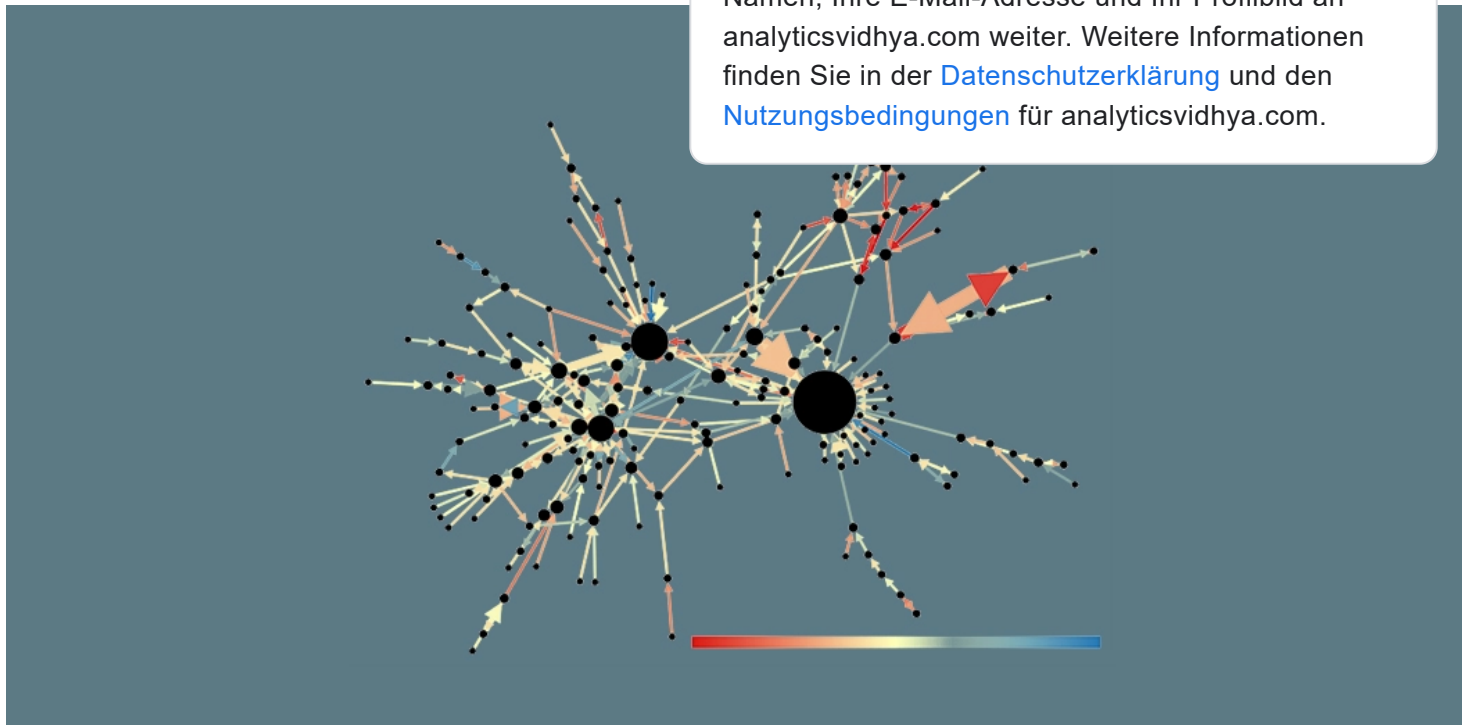
Article   🔒 Video Book   🔒 Interview Quiz

The Challenge of Making Machines Unde

"Language is a wonderful medium of communicatio

You and I would have understood that sentence in a   ss
text data in raw form. They need us to break down th   he
machine (the idea behind Natural Language Process
(https://courses.analyticsvidhya.com/courses/natu

(https://cdn.analyticsvidhya.com/wp-content/uploads/2020/02/word.jpg)

This is where the concepts of Bag-of-Words (BoW) and TF-IDF come into play. Both BoW and TF-IDF are
techniques that help us convert text sentences into numeric vectors.

I'll be discussing both Bag-of-Words and TF-IDF in this article. We'll use an intuitive and general example to
understand each concept in detail.

New to Natural Language Processing (NLP)? We've got the perfect courses for you to get started:

- *Introduction to Natural Language Processing (NLP) (https://courses.analyticsvidhya.com/courses/Intro-to-NLP) – Free course!*
- *Natural Language Processing (NLP) using Python (https://courses.analyticsvidhya.com/courses/natural-language-processing-nlp) – Comprehensive end-to-end NLP course*

Let's Take an Example to Understand Ba

I'll take a popular example to explain Bag-of-Words (

We all love watching movies (to varying degrees). I t
commit to watching it. I know a lot of you do the sar

Here's a sample of reviews about a particular horror movie:

- Review 1: This movie is very scary and long
- Review 2: This movie is not scary and is slow
- Review 3: This movie is spooky and good

You can see that there are some contrasting reviews about the movie as well as the length and pace of the movie. Imagine looking at a thousand reviews like these. Clearly, there is a lot of interesting insights we can draw from them and build upon them to gauge how well the movie performed.

However, as we saw above, we cannot simply give these sentences to a machine learning model and ask it to tell us whether a review was positive or negative. We need to perform certain text preprocessing steps.

Bag-of-Words and TF-IDF are two examples of how to do this. Let's understand them in detail.

Creating Vectors from Text

Can you think of some techniques we could use to v[...]
requirements would be:

1. It should not result in a sparse matrix since sp[...]
2. We should be able to retain most of the linguis[...]

Word Embedding (https://www.analyticsvidhya.com[...]
one such technique where we can represent the text [...]
embeddings are:

1. BoW, which stands for Bag of Words
2. TF-IDF, which stands for Term Frequency-Inverse Document Frequency

Now, let us see how we can represent the above movie reviews as embeddings and get them ready for a machine learning model.

Bag of Words (BoW) Model

The Bag of Words (BoW) model is the simplest form of text representation in numbers. Like the term itself, we can represent a sentence as a bag of words vector (a string of numbers).

Let's recall the three types of movie reviews we saw earlier:

- Review 1: This movie is very scary and long
- Review 2: This movie is not scary and is slow
- Review 3: This movie is spooky and good

We will first build a vocabulary from all the unique words in the above three reviews. The vocabulary consists of these 11 words: 'This', 'movie', 'is', 'very', 'scary', 'and', 'long', 'not', 'slow', 'spooky', 'good'.

We can now take each of these words and mark their occurrence in the three movie reviews above with 1s and 0s. This will give us 3 vectors for 3 reviews:

(https://cdn.analyticsvidhya.com/wp-content/uploa

Vector of Review 1: [1 1 1 1 1 1 1 0 0 0 0]

Vector of Review 2: [1 1 2 0 0 1 1 0 1 0 0]

Vector of Review 3: [1 1 1 0 0 0 1 0 0 1 1]

And that's the core idea behind a Bag of Words (BoW) model.

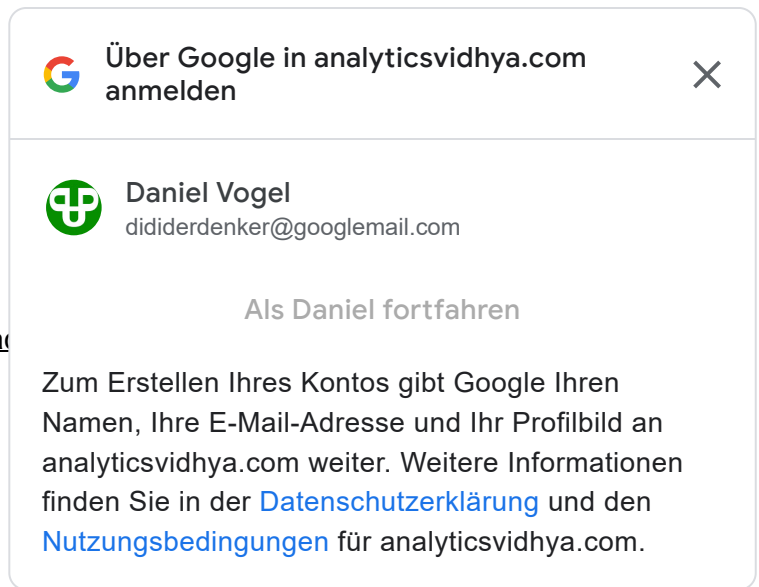## Drawbacks of using a Bag-of-Words (BoW) Model

In the above example, we can have vectors of length 11. However, we start facing issues when we come across new sentences:

1. If the new sentences contain new words, then our vocabulary size would increase and thereby, the length of the vectors would increase too.
2. Additionally, the vectors would also contain many 0s, thereby resulting in a sparse matrix (which is what we would like to avoid)
3. We are retaining no information on the grammar of the sentences nor on the ordering of the words in the text.

Term Frequency-Inverse Document Frequency (TF-IDF)

Let's first put a formal definition around TF-IDF. Here's how Wikipedia puts it:

> *"Term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus."*

Term Frequency (TF)

Let's first understand Term Frequent (TF). It is a mea...
d:

[(https://cdn.analyticsvidhya.com/wp-content/uploa...](https://cdn.analyticsvidhya.com/wp-content/uploa)

Here, in the numerator, n is the number of times the ... document and term would have its own TF value.

We will again use the same vocabulary we had built in the Bag-of-Words model to show how to calculate the TF for Review #2:

Review 2: This movie is not scary and is slow

Here,

- Vocabulary: 'This', 'movie', 'is', 'very', 'scary', 'and', 'long', 'not', 'slow', 'spooky', 'good'
- Number of words in Review 2 = 8
- TF for the word 'this' = (number of times 'this' appears in review 2)/(number of terms in review 2) = 1/8

Similarly,

- TF('movie') = 1/8
- TF('is') = 2/8 = 1/4
- TF('very') = 0/8 = 0
- TF('scary') = 1/8
- TF('and') = 1/8
- TF('long') = 0/8 = 0
- TF('not') = 1/8
- TF('slow') = 1/8
- TF('spooky') = 0/8 = 0
- TF('good') = 0/8 = 0

We can calculate the term frequencies for all the terms and all the reviews in this manner:
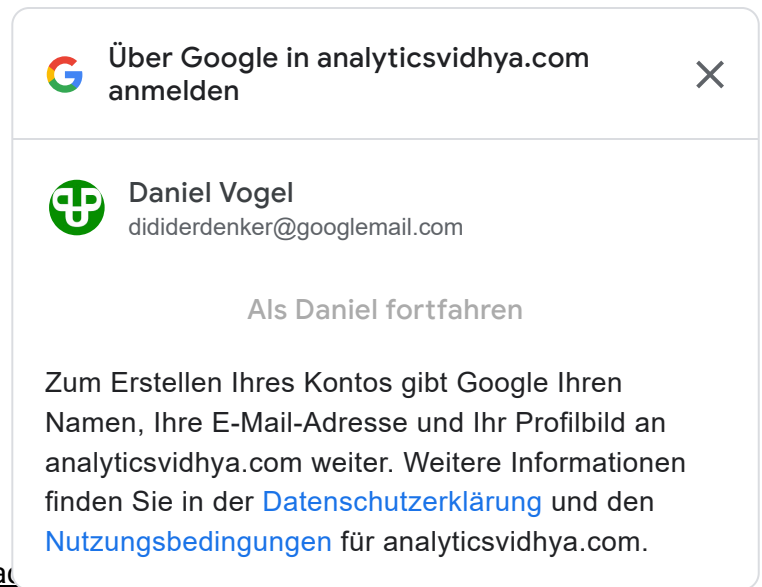
(https://cdn.analyticsvidhya.com/wp-content/upload

Inverse Document Frequency (IDF)

IDF is a measure of how important a term is. We need the IDF value because computing just the TF alone is not sufficient to understand the importance of words:

(https://cdn.analyticsvidhya.com/wp-content/uploads/2020/02/idf.jpg)

We can calculate the IDF values for the all the words in Review 2:
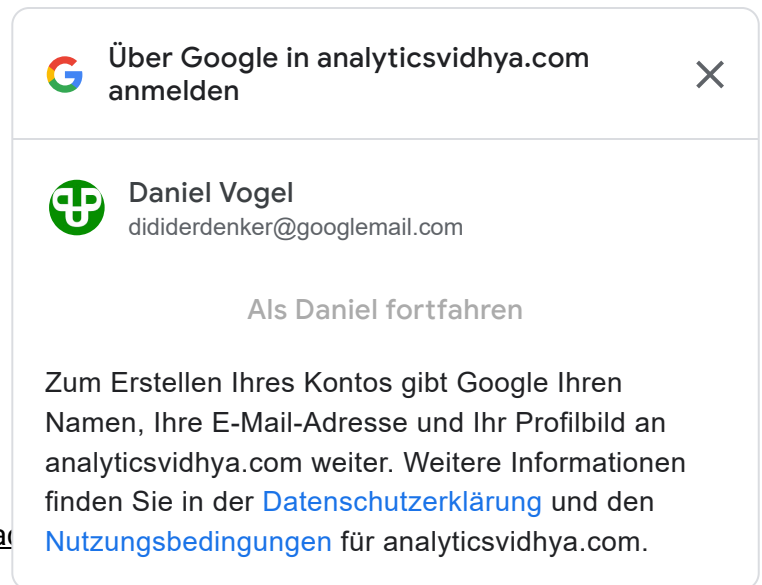
IDF('this') = log(number of documents/number of documents containing the word 'this') = log(3/3) = log(1) = 0

Similarly,

- IDF('movie',) = log(3/3) = 0
- IDF('is') = log(3/3) = 0
- IDF('not') = log(3/1) = log(3) = 0.48
- IDF('scary') = log(3/2) = 0.18
- IDF('and') = log(3/3) = 0
- IDF('slow') = log(3/1) = 0.48

We can calculate the IDF values for each word like this. Thus, the IDF values for the entire vocabulary would be:

(https://cdn.analyticsvidhya.com/wp-content/upload

Hence, we see that words like "is", "this", "and", etc., are reduced to 0 and have little importance; while words like "scary", "long", "good", etc. are words with more importance and thus have a higher value.

We can now compute the TF-IDF score for each word in the corpus. Words with a higher score are more important, and those with a lower score are less important:

(https://cdn.analyticsvidhya.com/wp-content/uploads/2020/02/tf_idf.jpg)

We can now calculate the TF-IDF score for every word in Review 2:

TF-IDF('this', Review 2) = TF('this', Review 2) * IDF('this') = 1/8 * 0 = 0

Similarly,

- TF-IDF('movie', Review 2) = 1/8 * 0 = 0
- TF-IDF('is', Review 2) = 1/4 * 0 = 0
- TF-IDF('not', Review 2) = 1/8 * 0.48 = 0.06
- TF-IDF('scary', Review 2) = 1/8 * 0.18 = 0.023
- TF-IDF('and', Review 2) = 1/8 * 0 = 0
- TF-IDF('slow', Review 2) = 1/8 * 0.48 = 0.06

Similarly, we can calculate the TF-IDF scores for all the words with respect to all the reviews:

(https://cdn.analyticsvidhya.com/wp-content/upload
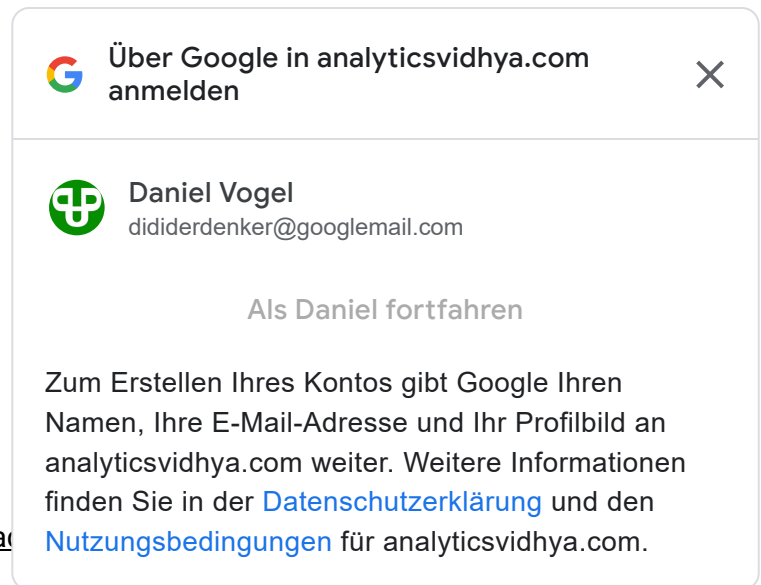
We have now obtained the TF-IDF scores for our vocabulary. TF-IDF also gives larger values for less frequent words and is high when both IDF and TF values are high i.e the word is rare in all the documents combined but frequent in a single document.

End Notes

Let me summarize what we've covered in the article:

1. Bag of Words just creates a set of vectors containing the count of word occurrences in the document (reviews), while the TF-IDF model contains information on the more important words and the less important ones as well.
2. Bag of Words vectors are easy to interpret. However, TF-IDF usually performs better in machine learning models.

While both Bag-of-Words and TF-IDF have been popular in their own regard, there still remained a void where understanding the context of words was concerned. Detecting the similarity between the words 'spooky' and 'scary', or translating our given documents into another language, requires a lot more information on the documents.

This is where Word Embedding techniques such as Word2Vec, Continuous Bag of Words (CBOW), Skipgram, etc. come in. You can find a detailed guide to such techniques here:

- An Intuitive Understanding of Word Embeddings: From Count Vectors to Word2Vec (https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/)

You can also read this article on our Mobile APP

(//play.google.com/store/apps/details?id=com.analyticsvidhya.android&utm_source=blog_global-all-co-prtnr-py-PartBadge-Mar2515-1)

(https://apps.apple.com/us/app/analytics-vidhya/id

Related Articles

(https://www.analyticsvidhya.com/blog/2017/01/ultimate-guide-to-understand-implement-natural-language-processing-codes-in-python/)
Ultimate Guide to Understand and Implement Natural Language Processing (with codes in Python) (https://www.analyticsvidhya.com/blog/2017/01/ultimate-guide-to-understand-implement-natural-language-processing-codes-in-python/)

(https://www.analyticsvidhya.com/blog/2018/02/the-different-methods-deal-text-data-predictive-python/)
Ultimate guide to deal with Text Data (using Python) - for Data Scientists and Engineers (https://www.analyticsvidhya.com/blog/2018/02/the-different-methods-deal-text-data-predictive-python/)

(https://www.analyticsvidhya.com/blog/2018/07/hands-on-sentiment-analysis-dataset-python/)
Comprehensive Hands on Guide to Twitter Sentiment Analysis with dataset and code (https://www.analyticsvidhya.com/blog/2018/07/hands-on-sentiment-analysis-dataset-python/)

TAGS : BEGINNER (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/BEGINNER/), MACHINE LEARNING (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/MACHINE-LEARNING/), NLP (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/NLP/), TEXT DATA (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/TEXT-DATA/), UNSTRUCTURED DATA (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/UNSTRUCTURED-DATA/)

NEX

A Beginner's Guide to matplotlib for D

(https://www.analyticsvidhya.com/blog/2020/02/beg

PREV

Joins in Pandas: Master the

(https://www.analyticsvidhya.com/blog/2020/02/jo

(https://www.analyticsvidhya.com/blog/author/purva91/)

Purva Huilgol (Https://Www.Analyticsvidhya.Com/Blog/Author/Purva91/)

Trainee Data Scientist at Analytics Vidhya. Pursuing Masters in Data Science from the University of Mumbai, Dept. of Computer Science. ML and NLP enthusiast.

This article is quite old and you might not get a prompt response from the author. We request you to post this comment on Analytics Vidhya's **Discussion portal** (https://discuss.analyticsvidhya.com/) to get your queries resolved

POPULAR POSTS

Commonly used Machine Learning Algorithms (w
(https://www.analyticsvidhya.com/blog/2017/09

Introductory guide on Linear Programming for (as
(https://www.analyticsvidhya.com/blog/2017/02
simple-english/)

6 Easy Steps to Learn Naive Bayes Algorithm witl
(https://www.analyticsvidhya.com/blog/2017/09

40 Questions to test a data scientist on Machine
DataFest 2017] (https://www.analyticsvidhya.cor   e-
learning-solution-skillpower-machine-learning-da

40 Questions to test a Data Scientist on Clusterin
(https://www.analyticsvidhya.com/blog/2017/02/test-data-scientist-clustering/)

45 Questions to test a data scientist on basics of Deep Learning (along with solution)
(https://www.analyticsvidhya.com/blog/2017/01/must-know-questions-deep-learning/)

30 Questions to test a data scientist on Linear Regression [Solution: Skilltest – Linear Regression]
(https://www.analyticsvidhya.com/blog/2017/07/30-questions-to-test-a-data-scientist-on-linear-
regression/)

Understanding Support Vector Machine(SVM) algorithm from examples (along with code)
(https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/)

CAREER RESOURCES

16 Key Questions You Should Answer Before Transitioning into Data
Science (https://www.analyticsvidhya.com/16-key-questions-data-
science-career-transition/?
&utm_source=Blog&utm_medium=CareerResourceWidget)
NOVEMBER 23, 2020

Here's What You Need to Know to Become a Data Scientist!
(https://www.analyticsvidhya.com/blog/2021/01/heres-what-you-
need-to-know-to-become-a-data-scientist/?
&utm_source=Blog&utm_medium=CareerResourceWidget)
JANUARY 22, 2021

These 7 Signs Show you have Data Scientist Potential!
(https://www.analyticsvidhya.com/blog/2020/12/these-7-signs-show-

you-have-data-scientist-potential/?
&utm_sourc

DECEMBER 3,

How To Hav
(https://ww
career-in-da
&utm_sourc

NOVEMBER 2

Should I bec
(https://ww
scientist-bu
&utm_sourc

NOVEMBER 24, 2020

RECENT POSTS

Beginners Guide to Robotics With Python (https://www.analyticsvidhya.com/blog/2021/03/beginners-guide-to-robotics-with-python/)

MARCH 23, 2021

A/B Testing Measurement Frameworks - Every Data Scientist Should Know (https://www.analyticsvidhya.com/blog/2021/03/a-b-testing-measurement-frameworks%e2%80%8a-%e2%80%8aevery-data-scientist-should-know/)

MARCH 23, 2021

Introduction to Image Segmentation for Data Science (https://www.analyticsvidhya.com/blog/2021/03/introduction-to-image-segmentation-for-data-science/)

MARCH 22, 2021

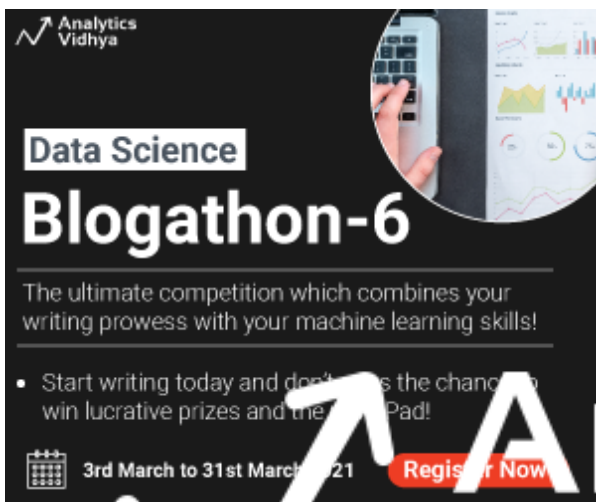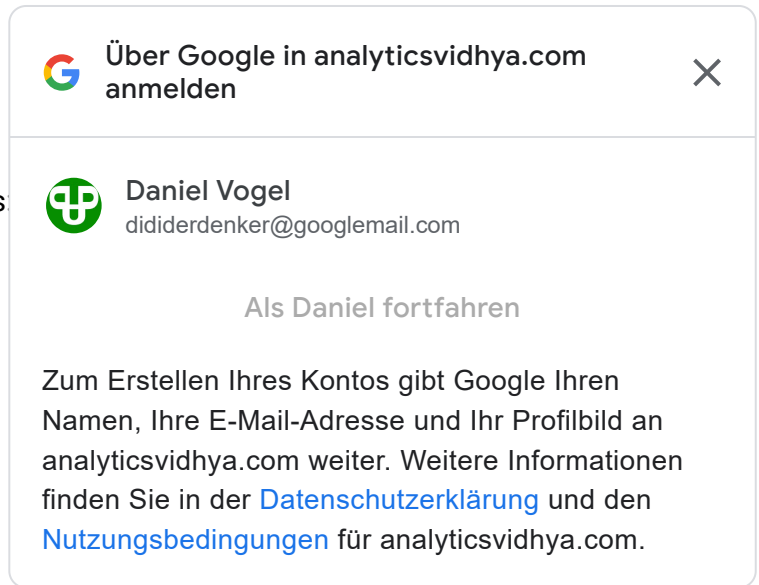**How to select Best Split in Decision trees using Gini Impurity (https://www.analyticsvidhya.com/blog/2021/03/how-to-select-best-split-in-decision-trees-gini-impurity/)**

MARCH 22, 2021

(https

utm_source=Blog&utm_medium=stickybanner1)

(https://datahack.analyticsvidhya.com/contest/data-science-

blogathon-6/?utm_source=blog&utm_medium=stickybanner2)

(https://www.analyticsvidhya.com/)

Download App

 (https://play.google.com/store/apps/details?id=com.analyticsvidhya.android)

 (https://apps.apple.com/us/app/analytics-vidhya/id1470025572)

Analytics Vidhya

About Us (https://www.analyticsvidhya.com/about-me/)

Our Team (https://www.analyticsvidhya.com/about-me/team/)

Careers (https://www.analyticsvidhya.com/about-me/career-analytics-vidhya/)

Data Science

Blog (https://www.analyticsvidhya.com/blog/)

Hackathon (https://datahack.analyticsvidhya.com/)

Discussions (https://discuss.analyticsvidhya.com/)

Apply Jobs (https://www.analyticsvidhya.com/jobs/)

Contact us (https://www.analyticsvidhya.com/contact/)

Companies

Visit us

in

f (https://www.linkedin.com/company/analytics-

(https://vidhya/facebook(https://youtwbetterscom/analytics.com/analytics-vidhya/UCHYagDteHtH4

© Copyright 201

Privacy Policy   Terms of Use   Refund Policy

Über Google in analyticsvidhya.com anmelden

Daniel Vogel
dididerdenker@googlemail.com

Als Daniel fortfahren

Zum Erstellen Ihres Kontos gibt Google Ihren Namen, Ihre E-Mail-Adresse und Ihr Profilbild an analyticsvidhya.com weiter. Weitere Informationen finden Sie in der Datenschutzerklärung und den Nutzungsbedingungen für analyticsvidhya.com.