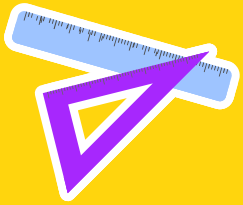







StackOverflow Tag Prediction

Garreth Cline

01

02

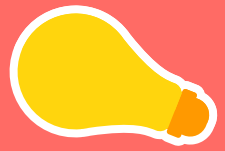
03

TABLE OF CONTENTS

01

02

03


01

BACKGROUND



A LITTLE BACKGROUND

TYPE OF SITE

QUESTION AND ANSWER BOARD

LOTS OF USERS

TEN MILLION

LOTS OF QUESTIONS

SIXTEEN MILLION

01

02

03

GOAL

Recommend tags for a question, based on title alone.

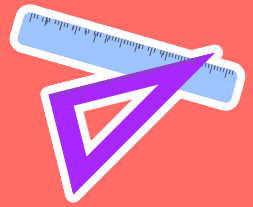
- Helps the user add more tags
- Add tags to a data set where tags are not available

01

02

03





The data I used was from a 400 GB MSSQL Server File

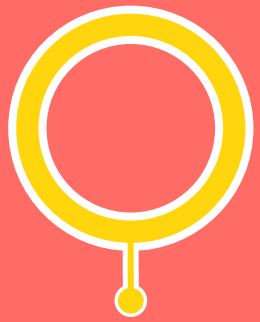


01

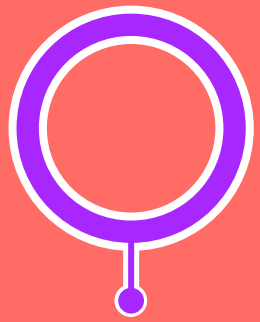
02

03

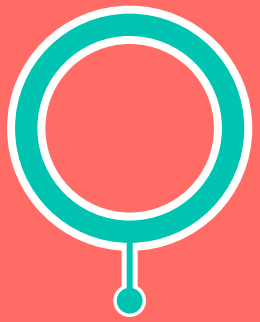
CONVERSION PATH



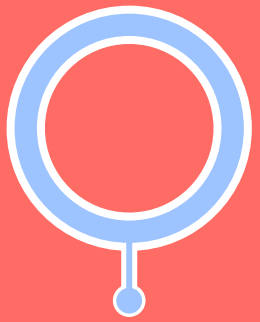
DATA DUMP



MSSQL



CSV



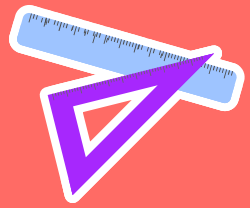
PANDAS

01

02

03

CLEANING THE DATA



Original

1000000

SCORE > 50

28183

View Count > 10000

27666

01

02

03

Features

Tag Features

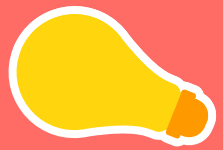
c# .net datetime

Each one is split and is on its own

Title

How do I calculate someone's age in C#?

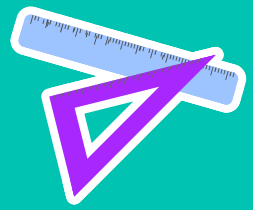
Used a TfidfVectorizer to make the words into useable data



01

02

03

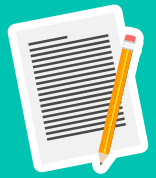


02

DATA



TAGS

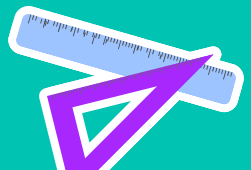


ALL TAGS

82,516

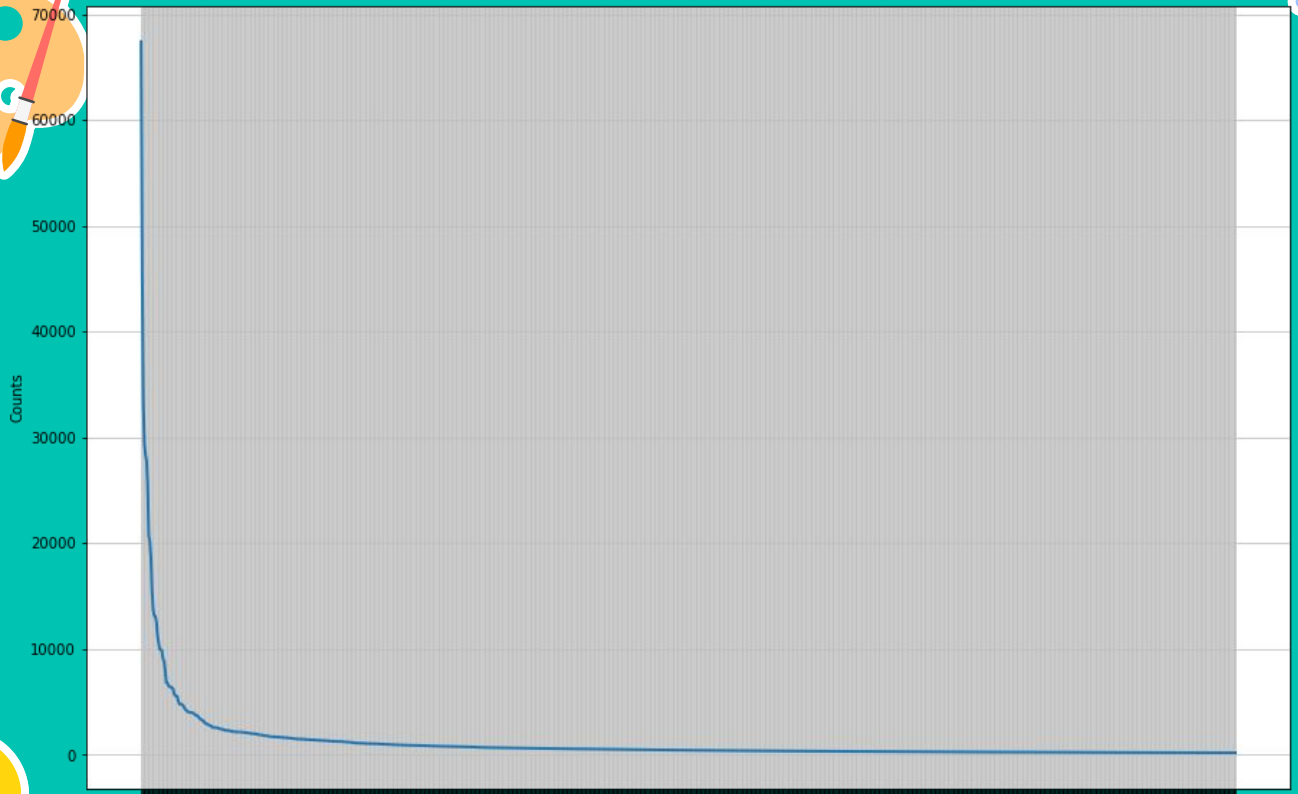
UNIQUE TAGS

6884



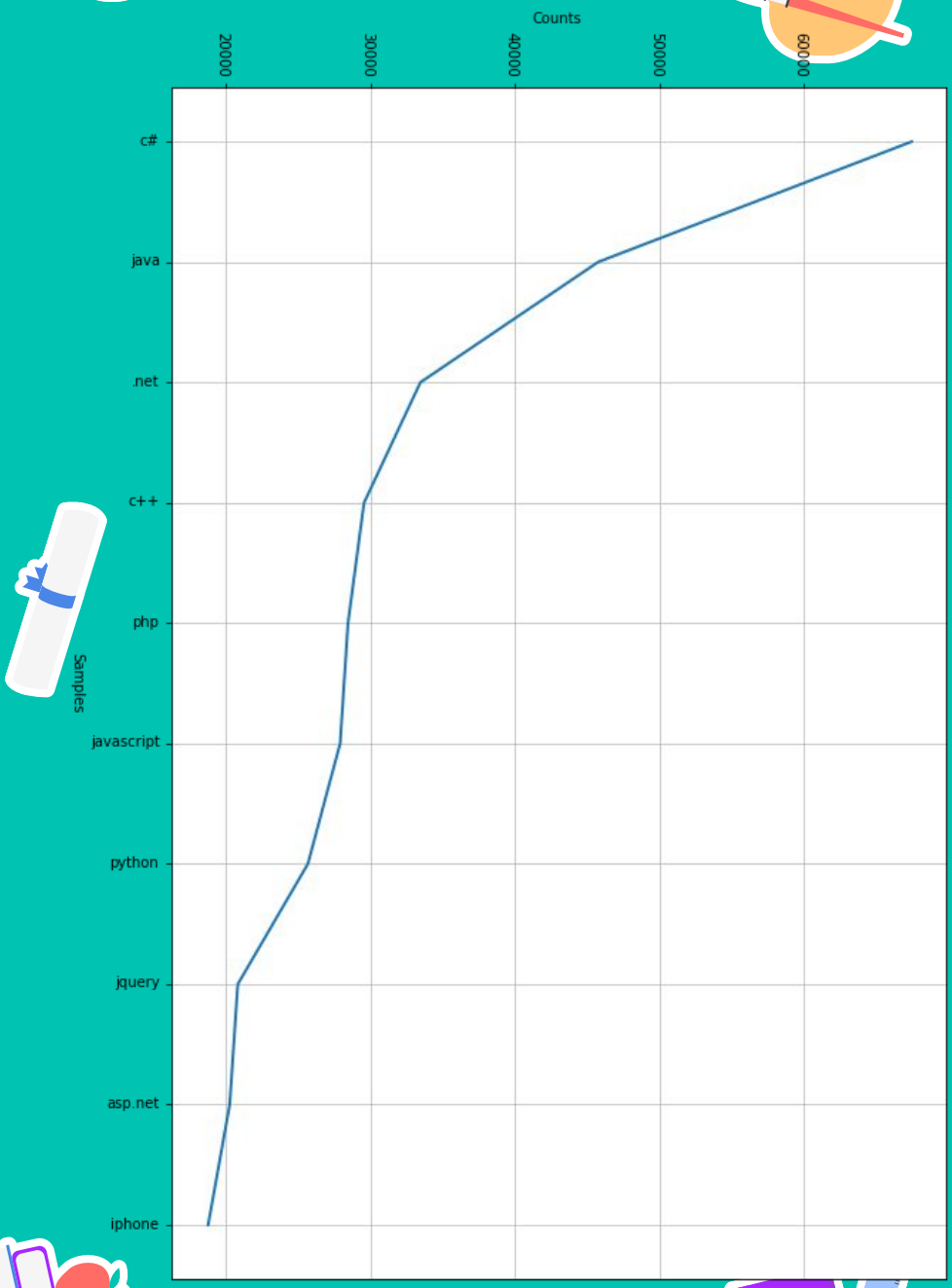
02

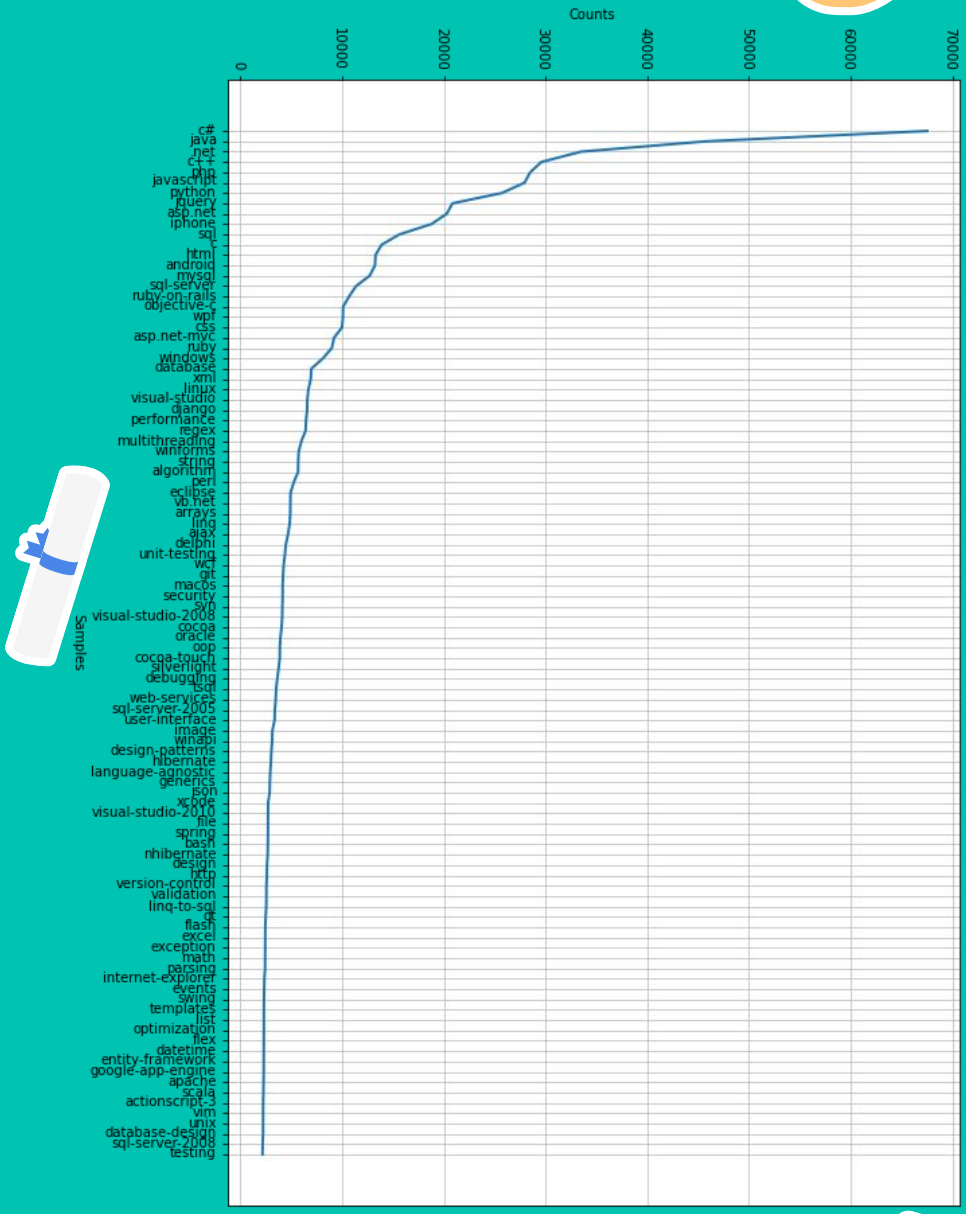
03



02

03


Counts
20000
30000
40000
50000
60000
Samples
c#
java
.net
c++
php
javascript
python
jquery
asp.net
iphone

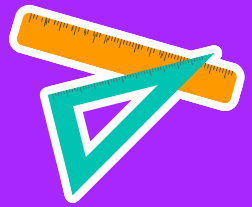

Counts
0
10000
20000
30000
40000
50000
60000
70000
Samples
c#
java
.net
c++
php
javascript
python
jquery
asp.net
iphone
sql
c
html
android
mysql
sql-server
ruby-on-rails
objective-c
wpf
css
asp.net-mvc
ruby
windows
database
xml
linux
visual-studio
django
performance
regex
multithreading
winforms
string
algorithm
perl
eclipse
vb.net
arrays
linq
ajax
delphi
unit-testing
wcf
git
macos
security
svn
visual-studio-2008
cocoa
oracle
oop
cocoa-touch
silverlight
debugging
tsql
web-services
sql-server-2005
user-interface
image
winapi
design-patterns
hibernate
language-agnostic
generics
json
xcode
visual-studio-2010
file
spring
bash
nhibernate
design
http
version-control
validation
linq-to-sql
qt
flash
excel
exception
math
parsing
internet-explorer
events
swing
templates
list
optimization
flex
datetime
entity-framework
google-app-engine
apache
scala
actionscript-3
vim
unix
database-design
sql-server-2008
testing



03

THE MODELS





Scores

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

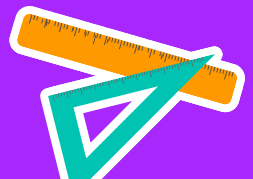
JACCARD SCORE

Jaccard similarity coefficient - size of the intersection divided by the size of the union of two label

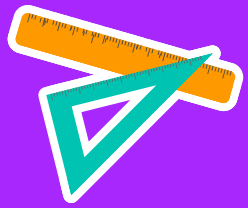
$$\frac{1}{|N| \cdot |L|} \sum_{i=1}^{|N|} \sum_{j=1}^{|L|} \operatorname{xor}(y_{i,j}, z_{i,j})$$

HAMMING LOSS

the fraction of the wrong labels to the total number of labels



DUMMY CLASSIFIER



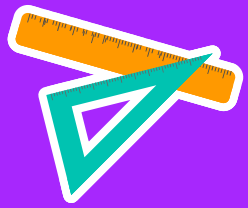
JACCARD SCORE

2.86

HAMMING LOSS

3.19

MultinomialNB (Naive Bayes)



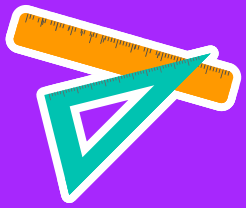
JACCARD SCORE

23.17

HAMMING LOSS

1.35

KNN



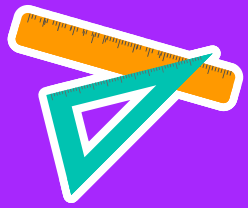
JACCARD SCORE

24.22

HAMMING LOSS

1.42

LogisticRegression



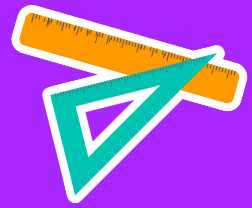
JACCARD SCORE

41.09

HAMMING LOSS

1.09

Perceptron (Linear Classifier)



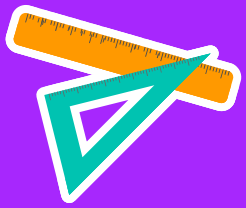
JACCARD SCORE

42.68

HAMMING LOSS

1.70

Linear Support Vector Classification



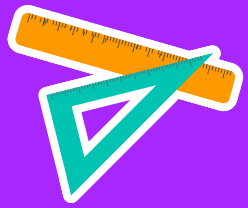
JACCARD SCORE

47.76

HAMMING LOSS

1.02

Decision Tree Classifier



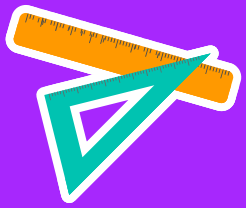
JACCARD SCORE

46.41

HAMMING LOSS

1.48

Random Forest



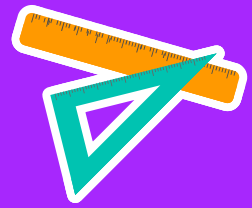
JACCARD SCORE

10 Trees - 44.60
100 Trees - 45.98
1000 Trees - 46.16

HAMMING LOSS

100 Trees - 1.08

Grid Search CV



Fitting 100 folds for each of 4 candidates, totalling 400 fits

JACCARD SCORE

47.76

HAMMING LOSS

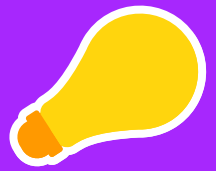
1.02

| Confusion Matrix 100 |

.net

[[4740 29]

[226 76]]



| Precision | Recall | F1 |

.net
|0.9347297554763214| 0.9457700650759219| 0.9369995044646904|

ajax
|0.9953924867742487| 0.995858804969434| 0.9950800654410777|

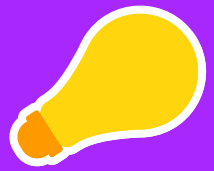
algorithm
|0.9873127809863937| 0.9891540130151844| 0.9864965925810498|

android
|0.9848270901232985| 0.985407217511339| 0.9846830805640743|

arrays
|0.9870868923706045| 0.9883652139617433| 0.9875110090286615|

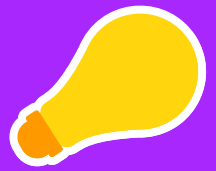
asp.net
|0.982017258455777| 0.9850128179846184| 0.9827719772503839|

asp.net-mvc
|0.9962843100981627| 0.9964504042595149| 0.9962259539020769|



| What title words are significant |

.net: 30 linq delegate nullable
wcf forms assembly 40 .net net



Conclusion

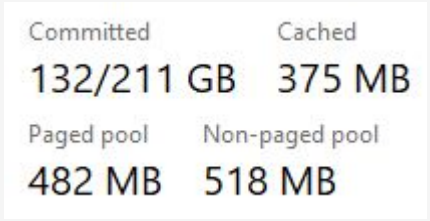
- Relatively high scores
- Predicts using just title data

CREDITS: This presentation template was created by **Slidesgo,** including icons by **Flaticon,** infographics & images by **Freepik**

Future

Scalability

- Formerly used cuDF and pyarrowDF
  - I relied on too many conversions it was not practical
  - Ran into problems due to my data being in unicode
- 80 20 Split
  - Took up 132 GBs or Ram
    - Used a huge paging file
  - Experimented with dask
    - Handles Spilling from GPU to CPU memory
      - Split the data into chunks and partitions that were useable
- Graphs are very busy and very intensive with 100 classes



Add body information

- Beautiful Soup
- Memory size is considerably larger
  - Decided to leave this out to have more rows

Any Questions?

.net
[[4740 29]
[226 76]]

.net
|0.9347297554763214| 0.9457700650759219| 0.9369995044646904|

ajax
|0.9953924867742487| 0.995858804969434| 0.9950800654410777|

algorithm
|0.9873127809863937| 0.9891540130151844| 0.9864965925810498|

android
|0.9848270901232985| 0.985407217511339| 0.9846830805640743|

arrays
|0.9870868923706045| 0.9883652139617433| 0.9875110090286615|

asp.net
|0.982017258455777| 0.9850128179846184| 0.9827719772503839|

asp.net-mvc
|0.9962843100981627| 0.9964504042595149| 0.9962259539020769|

.net: 30 linq delegate nullable
wcf forms assembly 40 .net net