Code Project
Spam Detection

Checkpoint One - Holocene Epoch

Kappagantula Lakshmi Abhigna (IMT2019040)
Vaibhavi Lokegaonkar (IMT2019090)

The Problem Statement

Create a model, which when given various details about a question and the account, can predict whether the question should be deleted, and also predict the reason for deletion (from a small set of standard reasons i.e multiclass classification).

The classes for the target are
0: "valid",
1: "kehna kya chahte ho"
2: "irrelevant"
3: "low quality"
4: "not reproducible"

Understanding the Data

Understanding the columns:

- **PostDateTime:** When the person has posted their question.
- **WhenAccountMade:** When the account was made
- **Heading:** The heading of the question
- **MainText:** The actual question (we realized that this included text + code)
- **PrimarySubject, SecondarySubject, TertiarySubject and OtherSubject:** Tags for each question (which subjects it pertains to)
- **Target:** The classification of the question as given in the actual problem statement

visual studio
access
link
button
one
rail
array
page
view
sql server
iphone
using
without
image
based
find
file
python
android
php
error
function
method
set
net mvc
add
change
possible
need
issue
getting
loop
list
string
table
asp
net
javascript
call
class
code
text
read
write
user
eclipse
check
create
state
java
problem
object
work
make
value
working
data
element
variable
ruby
difference

net mvc
asp net
objective c
sql server
javascript jquery
visual studio
c asp
net asp
iphone io
iphone objective
entity framework
internet explorer
html css
c net
rails ruby
java android
vb net
ruby
web service
php mysqljquery
jquery ajax
android android
sql sql
jquery jquery
c visual
google app
net c
mysql sql
rail
c window
iphone sdk
android layout
windows phone
c io
c code
c javascript
c spf
google chrome
jquery ui
jquery asp
ruby ruby
jquery plugin
jquery get
jquery event
jquery ajaxsubmit
jquery ajaxsubmit
jquery ajaxsubmit

Our Initial Observations :

- **'Karma' and 'NumAnswers':** The more the number of questions answered and total karma earned, the higher the probability that the person is an experienced user and will ask valid questions.

- **'PostDateTime' and 'WhenAccountMade':** They will help calculate the account age. The higher the account age, the longer the user has been on the platform.

- **Text columns:** We merged all the text columns to have the complete question along with the heading, text and tags.

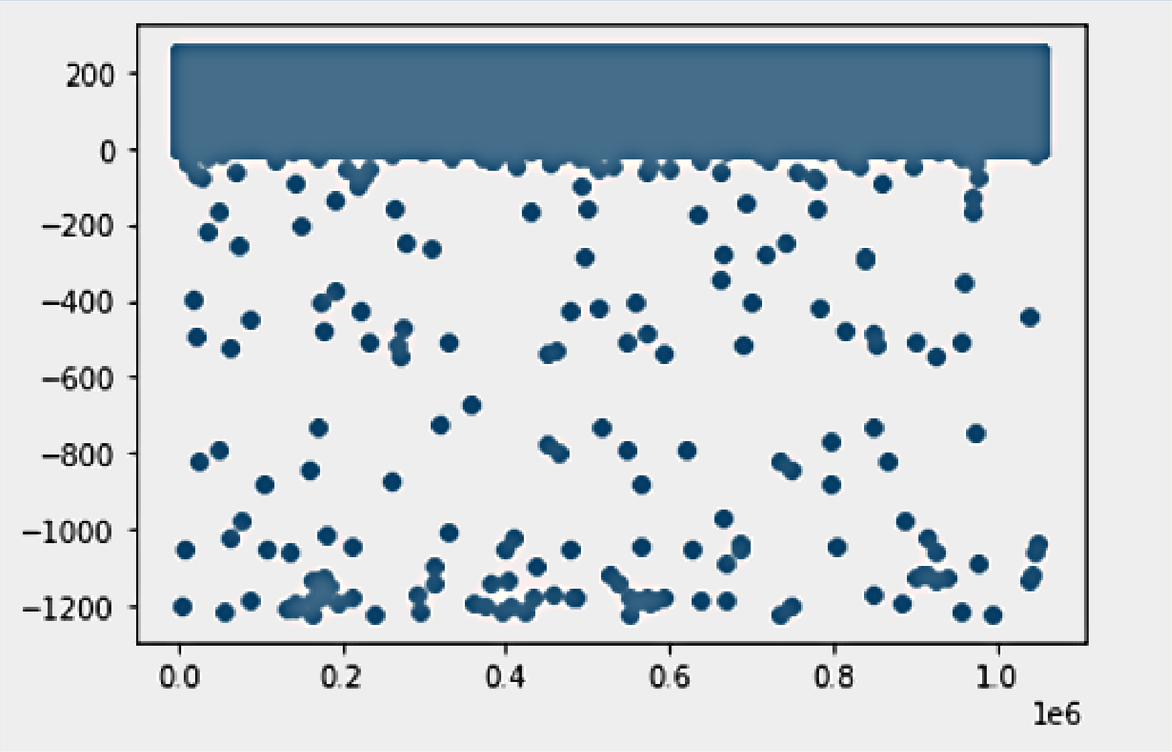
Step By Step Preprocessing

Step 1 - 'PostDateTime' and 'AccountDateTime' :

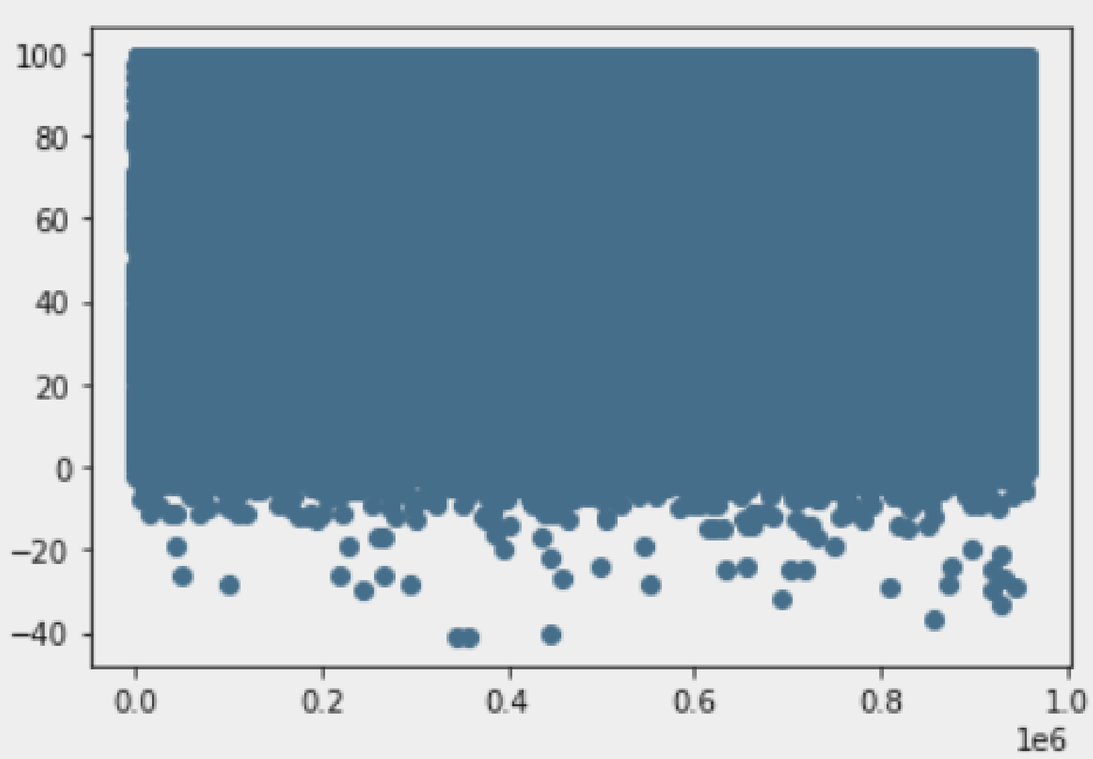
Non-uniform formats of the dates	Time was missing in some places	Creation of 'accountAge'
Converted all the dates and times to the format MM/DD/YY hh:mm:ss	Added 00:00:00 wherever time was missing	Converted all the dates and times to dateTime objects. Took the difference of the date time columns and dropped them to obtain 'accountAge'

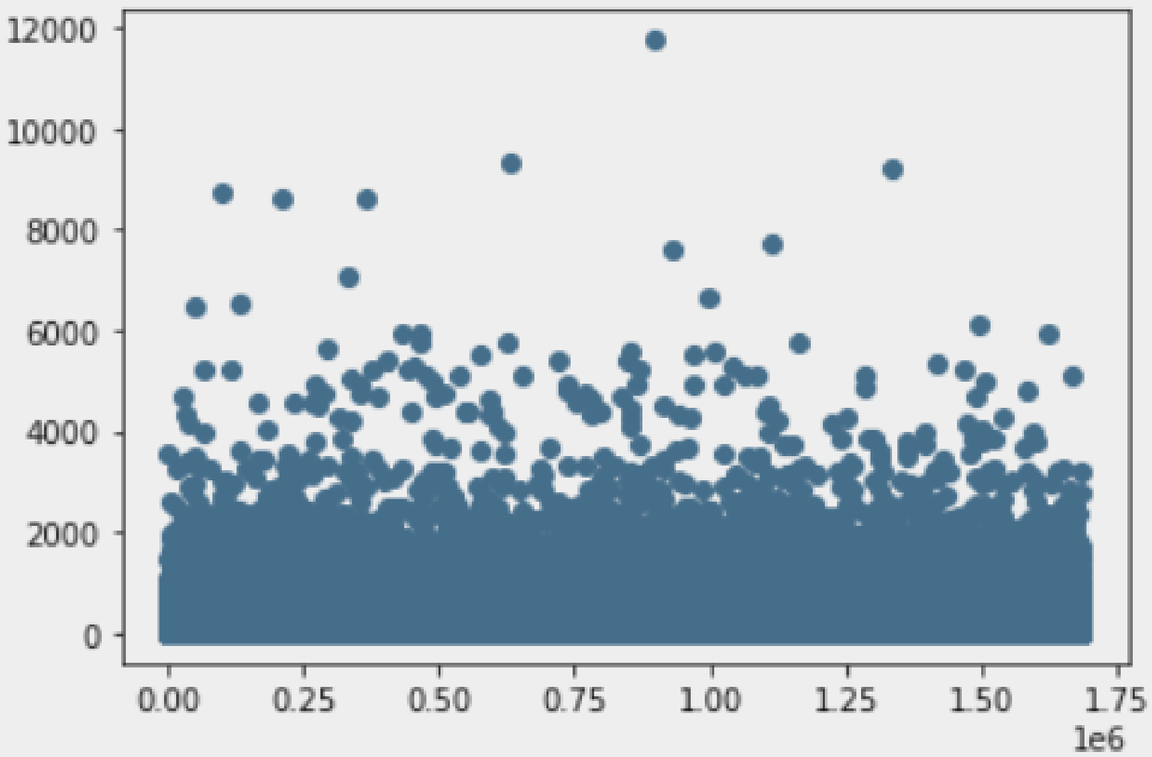
'accountAge'

Oh my god look at the negative values!



What about 'Karma' and 'NumAnswers'?





Karma has Negative Values which were problematic when we actually tried running a model.

Step 2: Deal with those

```
def make_col_positive(df, col):
    df[col] = df[col] + abs(df[col].min())
    return df
```

Step 3 - Nulls, Duplicates and 'OtherSubject' :

Nans	Duplicates	'OtherSubject' and other text columns with Nans
All the nulls values were present in the text columns	Dropped any duplicates	Replaced all Nans with '~'

Step 4 - Text Columns :

- We combined all of the text columns to create 'PostText'.
- 'Heading', 'MainText', 'PrimarySubject', 'SecondarySubject', 'TertiarySubject', 'OtherSubject' were dropped.
- 'Qid' was also dropped.

	Karma	NumAnswers	Target	accountAge	PostText
0	1	0	0	12	Populate dropdown list using ajax In grails we...
1	230	1	0	317	How to query database where date and time fiel...
2	1	0	0	83	Character Casing in Silverlight 4 TextBox I am...
3	5273	381	0	406	Stop exception from being thrown for non-exist...
4	8	0	0	14	search for campaigns with multiple values for ...

Corpus Preprocessing

Steps:

- Converted to lowercase.
- Removed special characters.
- Removed escape sequences.

Steps:

- Removed words of length less than or equal to 2.
- Removed stop words (used nltk, will try using gensim later)
- Stemming?

Generating the DTM

Memory Error	The Solution
After creating the corpus, our first idea was to try to convert the csr format into a dense format. We couldn't do that due to the sheer size of the data.	We retained the sparse format of the Document Term Matrix as obtained from the Count Vectorizer

What about the other columns?

We converted the rest of the columns - 'Karma', 'NumAnswers' and 'accountAge' into a csr. We then merged that csr with the DTM to obtain our final csr which got us ready to start trying the initial models.

Train Test Split:

- Randomized separation of the dataset into train and test sets might not give us all the classes because the data is highly skewed.
- Separation of the dataset needs to be done in such a way that the proportion of classes is the same in both train and test.
- This is ensured using the stratified parameter.

The Models

Multinomial Naive Bayes

Fast, but does not accept negative values

Accuracy was not very high

Logistic Regression

multi_class = 'ovr'
solver = 'liblinear'
class_weight = 'balanced'

Long runtime.

Accuracy higher as compared to Naive Bayes Models

Complement Naive Bayes

Fast

Accuracy was higher than Multinomial

What we plan on doing going forward?

We will go back and try more preprocessing methods to help make our models more efficient. We also plan on trying more models and improving on the ones we already tried. We will also look into models more suitable for larger datasets.

Thank You