wikiviv

December 7, 2016

1 Wikipedia Vandelism

1.1 Logistic Regression Model

In [1]:
```
import pandas as pd
from pandas import DataFrame, Series
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
edits = pd.read_csv('edits.csv')
annotators = pd.read_csv('annotators.csv')
annotations = pd.read_csv('annotations.csv')
gannotations = pd.read_csv('gold-annotations.csv')
```

In [3]:
```
edits.head()
```

Out[3]:

	editid	editor	oldrevisionid	newrevisionid	\
0	1	TheHeartbreakKid15	328391343	328391582	
1	2	Stepopen	327585467	327607921	
2	3	93.6.135.185	328227083	328242890	
3	4	Plasticspork	314955274	327191082	
4	5	Thatguyflint	329276563	329276581	

	diffurl	edittime
0	http://en.wikipedia.org/w/index.php?diff=32839...	2009-11-28T15:21:18Z
1	http://en.wikipedia.org/w/index.php?diff=32760...	2009-11-24T04:43:37Z
2	http://en.wikipedia.org/w/index.php?diff=32824...	2009-11-27T18:22:12Z
3	http://en.wikipedia.org/w/index.php?diff=32719...	2009-11-21T23:12:24Z
4	http://en.wikipedia.org/w/index.php?diff=32927...	2009-12-02T17:45:02Z

	editcomment	articleid	\
0	/* Episodes */	24477266	
1	removed factually wrong information	476288	
2	/* History */	174853	
3	Clean infobox + general fixes using [[Project:...	1418363	
4	Reverted edits by [[Special:Contributions/151...	1930796	

```
                                              articletitle
0                                 Top Gear (series 14)
1  List of United Nations resolutions concerning ...
2                                            W.A.S.P.
3                                         Psusennes II
4                               James W. Robinson, Jr.
```

```
In [4]: annotators.head()
```

Out[4]:

	annotatorid	age	sex	reading	editing	vandalizing	noticing
0	1	23.0	male	daily	less	no	weekly
1	2	23.0	male	daily	less	no	less
2	3	26.0	male	daily	less	yes	monthly
3	4	30.0	male	daily	never	no	never
4	5	27.0	male	weekly	never	no	never

```
In [5]: annotations.head()
```

Out[5]:

	editid	annotatorid	class	decisiontime	submittime
0	1642	83	no	7755	Tue Mar 02 19:46:32 GMT 2010
1	1643	83	no	21713	Tue Mar 02 19:46:32 GMT 2010
2	1641	83	no	11653	Tue Mar 02 19:46:32 GMT 2010
3	1640	83	no	10387	Tue Mar 02 19:46:32 GMT 2010
4	1639	83	no	10776	Tue Mar 02 19:46:32 GMT 2010

```
In [6]: gannotations.head()
```

Out[6]:

	editid	class	annotators	totalannotators
0	1	regular	3	3
1	2	regular	10	18
2	3	regular	3	3
3	4	regular	3	3
4	5	regular	5	6

```
In [7]: # gannotations, annotations and edits has 'editid' column as a common colu
        # merge annotators and annotations with annotatorid, then,
        # merge the resulting dataframe with gnnotations and edits
```

```
In [8]: merge1 = pd.merge(left = annotators, right = annotations, left_on= 'annotat
        merge2 = pd.merge(left = gannotations, right = merge1, left_on = 'editid',
        merge3 = pd.merge(left = edits, right = merge2, left_on = 'editid', right_o
        merge3.head(2)
```

Out[8]:

	editid	editor	oldrevisionid	newrevisionid	\
0	1	TheHeartbreakKid15	328391343	328391582	
1	1	TheHeartbreakKid15	328391343	328391582	

	diffurl	edittime
0	http://en.wikipedia.org/w/index.php?diff=32839...	2009-11-28T15:21:18Z

```
1  http://en.wikipedia.org/w/index.php?diff=32839...  2009-11-28T15:21:18Z

      editcomment  articleid          articletitle  class_x  \
0  /* Episodes */   24477266  Top Gear (series 14)  regular
1  /* Episodes */   24477266  Top Gear (series 14)  regular

                ...               annotatorid   age   sex  reading editing
0               ...                       1.0  23.0  male    daily    less
1               ...                       2.0  23.0  male    daily    less

  vandalizing noticing class_y decisiontime                    submittime
0          no   weekly      no      43453.0  Sat Feb 27 03:41:24 GMT 2010
1          no     less      no      14156.0  Sat Feb 27 14:55:29 GMT 2010

[2 rows x 22 columns]
```

```
In [9]: merge3.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 142027 entries, 0 to 142026
Data columns (total 22 columns):
editid             142027 non-null int64
editor             142027 non-null object
oldrevisionid      142027 non-null int64
newrevisionid      142027 non-null int64
diffurl            142027 non-null object
edittime           142027 non-null object
editcomment        142027 non-null object
articleid          142027 non-null int64
articletitle       142027 non-null object
class_x            142027 non-null object
annotators         142027 non-null int64
totalannotators    142027 non-null int64
annotatorid        142021 non-null float64
age                140498 non-null float64
sex                137328 non-null object
reading            141963 non-null object
editing            141907 non-null object
vandalizing        140582 non-null object
noticing           141933 non-null object
class_y            142021 non-null object
decisiontime       142021 non-null float64
submittime         142021 non-null object
dtypes: float64(3), int64(6), object(13)
memory usage: 24.9+ MB
```

```
In [10]: len(merge3.sex.unique())
```

```
Out[10]: 3

In [11]: len(merge3.class_x.unique())

Out[11]: 2

In [12]: merge3.class_x.value_counts()

Out[12]: regular      118376
         vandalism     23651
         Name: class_x, dtype: int64

In [13]: merge3.articletitle.value_counts()

Out[13]: Deaths in 2009                                                    118
         2012 (film)                                                        72
         Gunpowder Plot                                                     59
         2009 Great Britain and Ireland floods                              50
         Louis Lesser                                                       50
         Joseph Vijay                                                       46
         Telephone (song)                                                   42
         2009 in film                                                       42
         The Cry of Love                                                    42
         Assassins Creed II                                                 40
         Murder of Meredith Kercher                                         40
         Dancing with the Stars (U.S. season 9)                             40
         Jerome Is the New Black                                            40
         Stronger with Each Tear                                            40
         Jedward                                                            40
         George Washington and slavery                                      39
         Climatic Research Unit email controversy                           38
         WMMS                                                               38
         FIFA 10                                                            38
         52nd Grammy Awards                                                 38
         Survivor Series (2009)                                             38
         Liga Deportiva Universitaria de Quito                              37
         Catholic Church                                                    37
         List of iCarly episodes                                            37
         List of stage names                                                36
         Wakefield                                                          36
         Maguindanao massacre                                               36
         Herman Van Rompuy                                                  36
         Rudolf-Harbig-Stadion                                              36
         Rated R (Rihanna album)                                            35
                                                                           ...
         Havana (disambiguation)                                             1
         List of National Park Service areas in Georgia (U.S state)          1
         Marshawn Lynch                                                      1
         2009 in downloadable songs for the Rock Band series                 1
```

```
1st Battalion 1st Marines                                    1
List of Secretaries General of ASEAN                         1
European city bike                                           1
Gadhimai                                                     1
Singapore Armed Forces FC                                    1
Renault Espace                                               1
Institute for Sustainable Energy                             1
The Problem Solvers                                          1
Gatling gun                                                  1
Edgar Smith (pitcher)                                        1
Meat is Murder (book)                                        1
Wojdal                                                       1
Alpha Centauri in fiction                                    1
Skiathos Island National Airport                             1
Driving under the influence                                  1
Mother Angelica                                              1
The Legacy (professional wrestling)                          1
Drapers Mill, Margate                                        1
Scroll                                                       1
Jay Feely                                                    1
Power Rangers: Wild Force                                    1
Nora Tschirner                                               1
DeMar DeRozan                                                1
Catie Curtis                                                 1
Hamdan Al Kamali                                             1
Anthony Randolph                                             1
Name: articletitle, dtype: int64
```

```
In [14]: merge3.shape
```

```
Out[14]: (142027, 22)
```

```
In [15]: # Making a copy of dataset to keep main merged dataset original
         wikivand = merge3.copy()
```

```
In [16]: wikivand.shape
```

```
Out[16]: (142027, 22)
```

```
In [17]: wikivand.dtypes
```

```
Out[17]: editid              int64
         editor             object
         oldrevisionid       int64
         newrevisionid       int64
         diffurl            object
         edittime           object
         editcomment        object
         articleid           int64
```

```
articletitle        object
class_x             object
annotators           int64
totalannotators      int64
annotatorid        float64
age                float64
sex                 object
reading             object
editing             object
vandalizing         object
noticing            object
class_y             object
decisiontime       float64
submittime          object
dtype: object
```

```
In [18]: wikivand.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 142027 entries, 0 to 142026
Data columns (total 22 columns):
editid             142027 non-null int64
editor             142027 non-null object
oldrevisionid      142027 non-null int64
newrevisionid      142027 non-null int64
diffurl            142027 non-null object
edittime           142027 non-null object
editcomment        142027 non-null object
articleid          142027 non-null int64
articletitle       142027 non-null object
class_x            142027 non-null object
annotators         142027 non-null int64
totalannotators    142027 non-null int64
annotatorid        142021 non-null float64
age                140498 non-null float64
sex                137328 non-null object
reading            141963 non-null object
editing            141907 non-null object
vandalizing        140582 non-null object
noticing           141933 non-null object
class_y            142021 non-null object
decisiontime       142021 non-null float64
submittime         142021 non-null object
dtypes: float64(3), int64(6), object(13)
memory usage: 24.9+ MB
```

```
In [19]: wikivand.columns
```

```
Out[19]: Index(['editid', 'editor', 'oldrevisionid', 'newrevisionid', 'diffurl',
                'edittime', 'editcomment', 'articleid', 'articletitle', 'class_x',
                'annotators', 'totalannotators', 'annotatorid', 'age', 'sex', 'read
                'editing', 'vandalizing', 'noticing', 'class_y', 'decisiontime',
                'submittime'],
               dtype='object')

In [20]: wikivand.columns.values

Out[20]: array(['editid', 'editor', 'oldrevisionid', 'newrevisionid', 'diffurl',
                'edittime', 'editcomment', 'articleid', 'articletitle', 'class_x',
                'annotators', 'totalannotators', 'annotatorid', 'age', 'sex',
                'reading', 'editing', 'vandalizing', 'noticing', 'class_y',
                'decisiontime', 'submittime'], dtype=object)

In [21]: # After observing all the columns:  I am going to take the following steps
         # 1. Omiting few value
         # 2. Omit  rows with Na and nan - Na's
         # 3.1 formatting edittime and submit time - both are time variables
         # 3.2Adding new time differnce column
         # 3.3 removing submittime and edittime column
         # 4. Now, map vadalising yes to True and No to False
         # 5. Label Binarizer  for 'class_x', class_y', 'sex', 'reading','editing',
         # 6. Removed old 'class_x', class_y', 'sex', 'reading','editing', 'noticin
         #7. split the dataset into X_train, X_test, y_train, y_test
         #8. fit the logistic Regression model
         # 9. check accuray

In [22]: # Omit -
         # editor - as we have id's for various editors and it's just the name
         #diffurl - it's url of the wiki page, it must not have any effect on vanda

In [23]:  wikivand = wikivand.drop(['editor','diffurl'], axis=1)

In [24]: # panda.DataFrame.dropna()  - drops all the rows with any null values
         wikivand = wikivand.dropna()

In [25]: wikivand.shape

Out[25]: (135405, 20)

In [26]: wikivand.submittime= pd.to_datetime(wikivand.submittime)
         wikivand.submittime.head(2)

Out[26]: 0   2010-02-27 03:41:24
         1   2010-02-27 14:55:29
         Name: submittime, dtype: datetime64[ns]

In [27]: wikivand.edittime= pd.to_datetime(wikivand.edittime)
         wikivand.edittime.head(2)
```

```
Out[27]: 0   2009-11-28 15:21:18
         1   2009-11-28 15:21:18
         Name: edittime, dtype: datetime64[ns]

In [28]: import datetime
         wikivand['time'] =wikivand.submittime - wikivand.edittime
         # Pandas timestamp differences returns a datetime.timedelta object. This c
         wikivand.time.astype('timedelta64[h]').head(2)

Out[28]: 0    2172.0
         1    2183.0
         Name: time, dtype: float64

In [29]: # Now, we can have a new column with differnce of submittime and edittime
         wikivand['time'] =(wikivand.submittime - wikivand.edittime).dt.days
         #wikivand.time = wikivand.time.dt.days

In [30]: wikivand.columns

Out[30]: Index(['editid', 'oldrevisionid', 'newrevisionid', 'edittime', 'editcommen
                'articleid', 'articletitle', 'class_x', 'annotators', 'totalannotat
                'annotatorid', 'age', 'sex', 'reading', 'editing', 'vandalizing',
                'noticing', 'class_y', 'decisiontime', 'submittime', 'time'],
               dtype='object')

In [31]: #Now, I can drop submittime and edittime
         wikivand = wikivand.drop(['edittime','submittime'], axis=1)

In [32]: wikivand.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 135405 entries, 0 to 142026
Data columns (total 19 columns):
editid             135405 non-null int64
oldrevisionid      135405 non-null int64
newrevisionid      135405 non-null int64
editcomment        135405 non-null object
articleid          135405 non-null int64
articletitle       135405 non-null object
class_x            135405 non-null object
annotators         135405 non-null int64
totalannotators    135405 non-null int64
annotatorid        135405 non-null float64
age                135405 non-null float64
sex                135405 non-null object
reading            135405 non-null object
editing            135405 non-null object
vandalizing        135405 non-null object
noticing           135405 non-null object
```

```
class_y            135405 non-null object
decisiontime       135405 non-null float64
time               135405 non-null int64
dtypes: float64(3), int64(7), object(9)
memory usage: 20.7+ MB
```

```
In [33]: len(wikivand.noticing.unique())
```

```
Out[33]: 5
```

```
In [34]: wikivand.shape
```

```
Out[34]: (135405, 19)
```

```
In [35]: # mapping yes/ no to vandalizing to True/False
         d = {'yes': 1, 'no': 0};
         wikivand['vandalizing']=wikivand['vandalizing'].map(d);
```

```
In [36]: D = [{'foo': 1, 'bar': 2}, {'foo': 3, 'baz': 1}]
         D
```

```
Out[36]: [{'bar': 2, 'foo': 1}, {'baz': 1, 'foo': 3}]
```

```
In [37]: #Label Binarizer for class_y
         #a = wikivand[['class_y']]
         from sklearn import preprocessing
         lb = preprocessing.LabelBinarizer()
         temp1 = lb.fit_transform(wikivand[['class_y']])
         temp1 = pd.DataFrame(temp1, columns = [('class_y'+"_"+str(i)) for i in wik
         temp1 = temp1.set_index(wikivand.index.values)
         wikivand = pd.concat([wikivand, temp1], axis = 1)
         wikivand.head(2)
```

```
Out[37]:    editid  oldrevisionid  newrevisionid     editcomment  articleid  \
         0       1      328391343      328391582  /* Episodes */   24477266
         1       1      328391343      328391582  /* Episodes */   24477266

                    articletitle  class_x  annotators  totalannotators  annotatoric
         0  Top Gear (series 14)  regular           3                3          1.0
         1  Top Gear (series 14)  regular           3                3          2.0

                ...        editing vandalizing noticing class_y  decisiontime time
         0      ...           less           0   weekly      no       43453.0   90
         1      ...           less           0     less      no       14156.0   90

           class_y_no  class_y_yes  class_y_dunno  class_y_error
         0          0            0              1              0
         1          0            0              1              0

         [2 rows x 23 columns]
```

In [38]:
```
# . Label Binarizer  for 'class_x'
from sklearn import preprocessing
lb = preprocessing.LabelBinarizer()
temp2 = lb.fit_transform(wikivand[['class_x']])
temp2 = pd.DataFrame(temp2)
temp2.columns = ['class_x_lb']
temp2 = temp2.set_index(wikivand.index.values)
wikivand = pd.concat([wikivand, temp2], axis = 1)
wikivand.head(2)
```

Out[38]:

	editid	oldrevisionid	newrevisionid	editcomment	articleid	\
0	1	328391343	328391582	/* Episodes */	24477266	
1	1	328391343	328391582	/* Episodes */	24477266	

	articletitle	class_x	annotators	totalannotators	annotatoric
0	Top Gear (series 14)	regular	3	3	1.0
1	Top Gear (series 14)	regular	3	3	2.0

	...	vandalizing	noticing	class_y	decisiontime	time	class_y_no
0	...	0	weekly	no	43453.0	90	0
1	...	0	less	no	14156.0	90	0

	class_y_yes	class_y_dunno	class_y_error	class_x_lb
0	0	1	0	0
1	0	1	0	0

[2 rows x 24 columns]

In [39]:
```
# . Label Binarizer  for 'sex',
from sklearn import preprocessing
lb = preprocessing.LabelBinarizer()
temp3 = lb.fit_transform(wikivand[['sex']])
temp3 = pd.DataFrame(temp3)
temp3.columns = ['sex_lb']
temp3 = temp3.set_index(wikivand.index.values)
wikivand = pd.concat([wikivand, temp3], axis = 1)
wikivand.head(2)
```

Out[39]:

	editid	oldrevisionid	newrevisionid	editcomment	articleid	\
0	1	328391343	328391582	/* Episodes */	24477266	
1	1	328391343	328391582	/* Episodes */	24477266	

	articletitle	class_x	annotators	totalannotators	annotatoric
0	Top Gear (series 14)	regular	3	3	1.0
1	Top Gear (series 14)	regular	3	3	2.0

	...	noticing	class_y	decisiontime	time	class_y_no	class_y_yes	\
0	...	weekly	no	43453.0	90	0	0	

```
1   ...        less      no      14156.0   90           0           0

  class_y_dunno  class_y_error  class_x_lb  sex_lb
0             1              0           0       1
1             1              0           0       1

[2 rows x 25 columns]
```

In [40]:
```
# . Label Binarizer  for  'reading'
from sklearn import preprocessing
lb = preprocessing.LabelBinarizer()
temp4 = lb.fit_transform(wikivand[['reading']])
temp4 = pd.DataFrame(temp4, columns = [('reading'+"_"+str(i)) for i in wik
temp4 = temp4.set_index(wikivand.index.values)
wikivand = pd.concat([wikivand, temp4], axis = 1)
wikivand.head(2)
```

Out[40]:
```
   editid  oldrevisionid  newrevisionid     editcomment  articleid  \
0       1      328391343      328391582  /* Episodes */   24477266
1       1      328391343      328391582  /* Episodes */   24477266

           articletitle  class_x  annotators  totalannotators  annotatoric
0  Top Gear (series 14)  regular           3                3          1.0
1  Top Gear (series 14)  regular           3                3          2.0

       ...       class_y_no class_y_yes class_y_dunno class_y_error  \
0      ...                0           0             1             0
1      ...                0           0             1             0

   class_x_lb sex_lb reading_daily  reading_weekly  reading_monthly  \
0           0      1             1               0                0
1           0      1             1               0                0

   reading_less
0             0
1             0

[2 rows x 29 columns]
```

In [41]:
```
# . Label Binarizer  for editing
from sklearn import preprocessing
lb = preprocessing.LabelBinarizer()
temp5 = lb.fit_transform(wikivand[['editing']])
temp5 = pd.DataFrame(temp5, columns = [('editing'+"_"+str(i)) for i in wik
temp5 = temp5.set_index(wikivand.index.values)
wikivand = pd.concat([wikivand, temp5], axis = 1)
wikivand.head(2)
```

Out[41]:
```
   editid  oldrevisionid  newrevisionid     editcomment  articleid  \
0       1      328391343      328391582  /* Episodes */   24477266
```

```
1       1      328391343      328391582  /* Episodes */   24477266

           articletitle  class_x  annotators  totalannotators  annotatoric
0  Top Gear (series 14)  regular           3                3          1.0
1  Top Gear (series 14)  regular           3                3          2.0

       ...        sex_lb reading_daily reading_weekly reading_monthly  \
0      ...             1             1              0               0
1      ...             1             1              0               0

   reading_less editing_never editing_less  editing_monthly  editing_weekl
0             0             0            1                0
1             0             0            1                0

   editing_daily
0              0
1              0

[2 rows x 34 columns]
```

In [42]:
```
# . Label Binarizer  for noticing
from sklearn import preprocessing
lb = preprocessing.LabelBinarizer()
temp6 = lb.fit_transform(wikivand[['noticing']])
temp6 = pd.DataFrame(temp6, columns = [('noticing'+"_"+str(i)) for i in wi
temp6 = temp6.set_index(wikivand.index.values)
wikivand = pd.concat([wikivand, temp6], axis = 1)
wikivand.head(2)
```

Out[42]:
```
   editid  oldrevisionid  newrevisionid     editcomment  articleid  \
0       1      328391343      328391582  /* Episodes */   24477266
1       1      328391343      328391582  /* Episodes */   24477266

           articletitle  class_x  annotators  totalannotators  annotatoric
0  Top Gear (series 14)  regular           3                3          1.0
1  Top Gear (series 14)  regular           3                3          2.0

       ...        editing_never editing_less editing_monthly editing_week
0      ...                    0            1               0
1      ...                    0            1               0

   editing_daily noticing_less noticing_monthly  noticing_never  \
0              0             0                0               0
1              0             0                1               0

   noticing_weekly  noticing_daily
0                0               1
1                0               0
```

```
         [2 rows x 39 columns]
```

```
In [43]: wikivand.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 135405 entries, 0 to 142026
Data columns (total 39 columns):
editid              135405 non-null int64
oldrevisionid       135405 non-null int64
newrevisionid       135405 non-null int64
editcomment         135405 non-null object
articleid           135405 non-null int64
articletitle        135405 non-null object
class_x             135405 non-null object
annotators          135405 non-null int64
totalannotators     135405 non-null int64
annotatorid         135405 non-null float64
age                 135405 non-null float64
sex                 135405 non-null object
reading             135405 non-null object
editing             135405 non-null object
vandalizing         135405 non-null int64
noticing            135405 non-null object
class_y             135405 non-null object
decisiontime        135405 non-null float64
time                135405 non-null int64
class_y_no          135405 non-null int64
class_y_yes         135405 non-null int64
class_y_dunno       135405 non-null int64
class_y_error       135405 non-null int64
class_x_lb          135405 non-null int64
sex_lb              135405 non-null int64
reading_daily       135405 non-null int64
reading_weekly      135405 non-null int64
reading_monthly     135405 non-null int64
reading_less        135405 non-null int64
editing_never       135405 non-null int64
editing_less        135405 non-null int64
editing_monthly     135405 non-null int64
editing_weekly      135405 non-null int64
editing_daily       135405 non-null int64
noticing_less       135405 non-null int64
noticing_monthly    135405 non-null int64
noticing_never      135405 non-null int64
noticing_weekly     135405 non-null int64
noticing_daily      135405 non-null int64
dtypes: float64(3), int64(28), object(8)
```

```
memory usage: 41.3+ MB
```

In [44]:
```
# Now I need to drop the old  ['class_x', 'class_y', 'sex', 'reading','edi
wikivand = wikivand.drop(['class_x', 'class_y', 'sex', 'reading','editing'
wikivand.head(2)
```

Out[44]:

	editid	oldrevisionid	newrevisionid	editcomment	articleid	\
0	1	328391343	328391582	/* Episodes */	24477266	
1	1	328391343	328391582	/* Episodes */	24477266	

	articletitle	annotators	totalannotators	annotatorid	age	\
0	Top Gear (series 14)	3	3	1.0	23.0	
1	Top Gear (series 14)	3	3	2.0	23.0	

	...	editing_never	editing_less	editing_monthly	\
0	...	0	1	0	
1	...	0	1	0	

	editing_weekly	editing_daily	noticing_less	noticing_monthly	\
0	0	0	0	0	
1	0	0	0	1	

	noticing_never	noticing_weekly	noticing_daily
0	0	0	1
1	0	0	0

[2 rows x 33 columns]

In [45]:
```
# wikivand has all numeric features and two string features: comments and
```

In [46]:
```
# Let's study all the features other than these two
wikivand1 = wikivand.copy()
wikivand1.head(2)
```

Out[46]:

	editid	oldrevisionid	newrevisionid	editcomment	articleid	\
0	1	328391343	328391582	/* Episodes */	24477266	
1	1	328391343	328391582	/* Episodes */	24477266	

	articletitle	annotators	totalannotators	annotatorid	age	\
0	Top Gear (series 14)	3	3	1.0	23.0	
1	Top Gear (series 14)	3	3	2.0	23.0	

	...	editing_never	editing_less	editing_monthly	\
0	...	0	1	0	
1	...	0	1	0	

	editing_weekly	editing_daily	noticing_less	noticing_monthly	\
0	0	0	0	0	

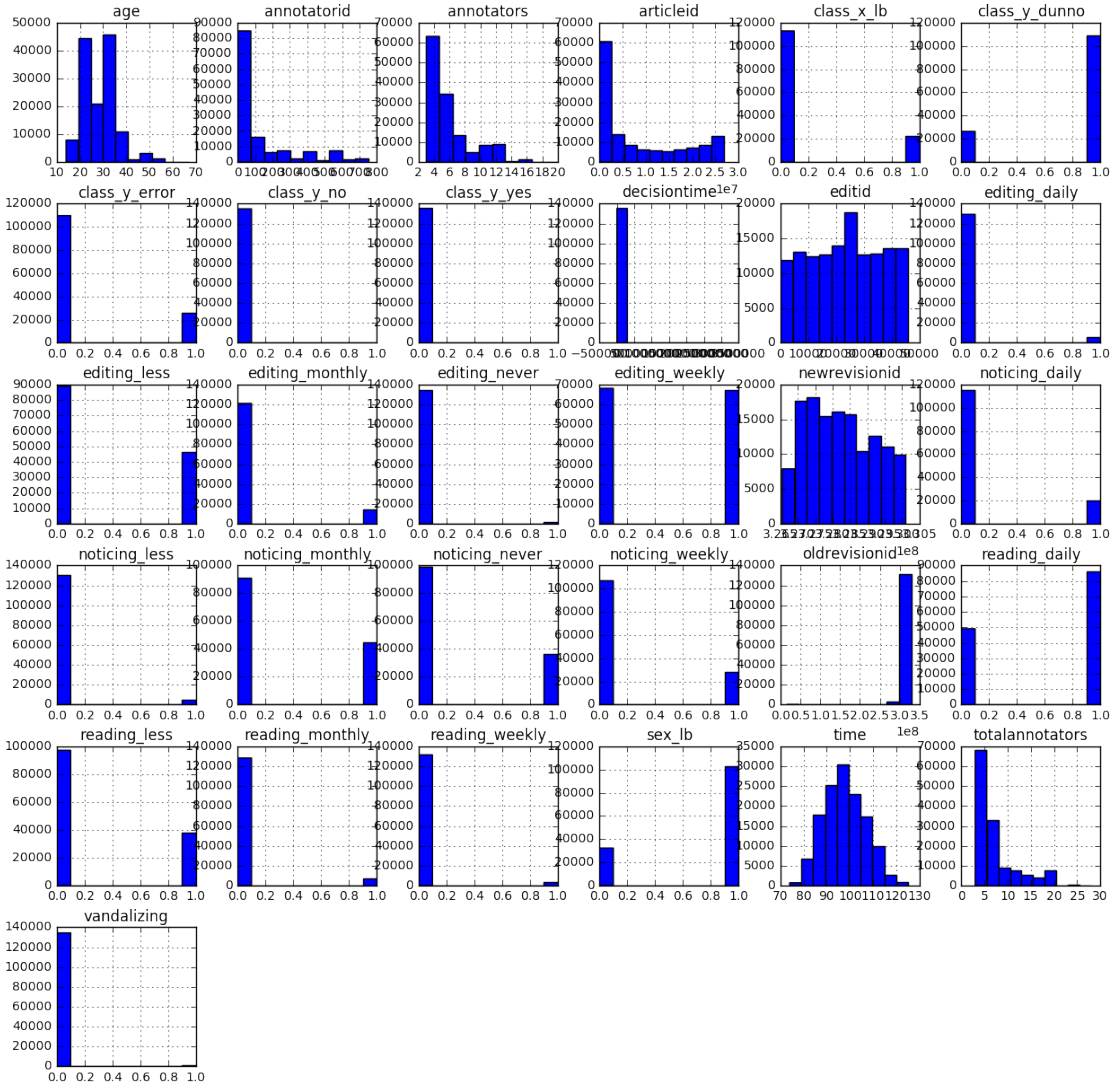
```
1                0                0                0                    1

   noticing_never  noticing_weekly  noticing_daily
0                0                0               1
1                0                0               0

[2 rows x 33 columns]
```

In [47]:
```
# Let's plot the distribution of all the continuous variables in  wikivand
```

In [48]:
```
import matplotlib.pyplot as plt
%matplotlib inline
plot1= wikivand1[wikivand1.dtypes[(wikivand1.dtypes=="float64")|(wikivand1
```



```
In [53]: wikivand1 = wikivand1[wikivand1.dtypes[(wikivand1.dtypes=="float64")|(wiki
         wikivand1.head(2)  # A dataframe with only int and float values

Out[53]:    editid  oldrevisionid  newrevisionid  articleid  annotators  \
         0       1      328391343      328391582   24477266           3
         1       1      328391343      328391582   24477266           3

            totalannotators  annotatorid   age  vandalizing  decisiontime  \
         0                3          1.0  23.0            0       43453.0
         1                3          2.0  23.0            0       14156.0

                ...        editing_never  editing_less  editing_monthly  \
         0      ...                    0             1                0
         1      ...                    0             1                0

            editing_weekly  editing_daily  noticing_less  noticing_monthly  \
         0               0              0              0                 0
         1               0              0              0                 1

            noticing_never  noticing_weekly  noticing_daily
         0               0                0               1
         1               0                0               0

         [2 rows x 31 columns]

In [54]: y = wikivand1.vandalizing.astype(int)
         y.head(2)

Out[54]: 0    0
         1    0
         Name: vandalizing, dtype: int64

In [55]: x = wikivand1.drop(['vandalizing'], axis = 1)
         x.head(2)

Out[55]:    editid  oldrevisionid  newrevisionid  articleid  annotators  \
         0       1      328391343      328391582   24477266           3
         1       1      328391343      328391582   24477266           3

            totalannotators  annotatorid   age  decisiontime  time       ...
         0                3          1.0  23.0       43453.0    90       ...
         1                3          2.0  23.0       14156.0    90       ...

            editing_never  editing_less  editing_monthly  editing_weekly  \
         0              0             1                0               0
         1              0             1                0               0

            editing_daily  noticing_less  noticing_monthly  noticing_never  \
         0              0              0                 0               0
```

```
1              0              0                  1                0

   noticing_weekly  noticing_daily
0                0               1
1                0               0

[2 rows x 30 columns]
```

```
In [56]: # Defining x_train, y_train, x_test, y_test
         from sklearn.cross_validation import train_test_split
         x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, r
```

```
In [59]: from sklearn.linear_model import LogisticRegression
         from sklearn import datasets
         from sklearn import metrics
         model1 = LogisticRegression(class_weight = 'balanced')
         model1.fit(x_train,y_train)

Out[59]: LogisticRegression(C=1.0, class_weight='balanced', dual=False,
                   fit_intercept=True, intercept_scaling=1, max_iter=100,
                   multi_class='ovr', n_jobs=1, penalty='l2', random_state=None,
                   solver='liblinear', tol=0.0001, verbose=0, warm_start=False)
```

```
In [60]: # make predictions
         expected = y_test
         predicted = model1.predict(x_test)
         # summarize the fit of the model
         print(metrics.classification_report(expected, predicted))

             precision    recall  f1-score   support

          0       0.99      0.72      0.84     26885
          1       0.01      0.49      0.02       196

avg / total       0.99      0.72      0.83     27081
```

```
In [61]: print(metrics.confusion_matrix(expected, predicted))

[[19389  7496]
 [  100    96]]
```

```
In [62]: model1.score(x_test, y_test)

Out[62]: 0.71950814223994686
```

```
In [67]: # Support Vector Machine

         from sklearn import metrics
         from sklearn.svm import SVC
         # fit a SVM model to the data
         model2 = SVC(class_weight = 'balanced')
         model2.fit(x_train, y_train)

Out[67]: SVC(C=1.0, cache_size=200, class_weight='balanced', coef0=0.0,
           decision_function_shape=None, degree=3, gamma='auto', kernel='rbf',
           max_iter=-1, probability=False, random_state=None, shrinking=True,
           tol=0.001, verbose=False)

In [68]: # make predictions
         expected2 = y_test
         predicted2 = model2.predict(x_test)
         # summarize the fit of the model
         print(metrics.classification_report(expected2, predicted2))

                    precision    recall  f1-score   support

                 0       0.99      1.00      1.00     26885
                 1       0.00      0.00      0.00       196

         avg / total       0.99      0.99      0.99     27081


         /Users/Vivek/anaconda/lib/python3.5/site-packages/sklearn/metrics/classification.py
           'precision', 'predicted', average, warn_for)


In [69]: print(metrics.confusion_matrix(expected2, predicted2))

         [[26885     0]
          [  196     0]]


In [70]: model2.score(x_test, y_test)

Out[70]: 0.99276245338059899

In [71]: '''We are getting around 99%accuracy with both Logistic Regression and Sup
         May be the data is over fitting or editors id is more influencign and few
         Now, we will take a look at how comment influence out prediction accuracy.

Out[71]: 'We are getting around 99%accuracy with both Logistic Regression and Suppo

In [72]: # we will make second copy of wikivand for this
         wikivand2 = wikivand.copy()
         wikivand2.head(2)
```

Out[72]:

	editid	oldrevisionid	newrevisionid	editcomment	articleid	\
0	1	328391343	328391582	/* Episodes */	24477266	
1	1	328391343	328391582	/* Episodes */	24477266	

	articletitle	annotators	totalannotators	annotatorid	age	\
0	Top Gear (series 14)	3	3	1.0	23.0	
1	Top Gear (series 14)	3	3	2.0	23.0	

	...	editing_never	editing_less	editing_monthly	\
0	...	0	1	0	
1	...	0	1	0	

	editing_weekly	editing_daily	noticing_less	noticing_monthly	\
0	0	0	0	0	
1	0	0	0	1	

	noticing_never	noticing_weekly	noticing_daily
0	0	0	1
1	0	0	0

[2 rows x 33 columns]

In [73]:
```
# Vectorising EDITComments
from sklearn.feature_extraction.text import CountVectorizer
corpus = wikivand2['editcomment']
```

In [74]:
```
# Creating 1000 features from comments
vectorizer = CountVectorizer(min_df = 1, stop_words = 'english', max_featu
```

In [75]:
```
X = vectorizer.fit_transform(corpus)
X
```

Out[75]:
```
<135405x1000 sparse matrix of type '<class 'numpy.int64'>'
        with 361706 stored elements in Compressed Sparse Row format>
```

In [76]:
```
X.toarray
```

Out[76]:
```
<bound method _cs_matrix.toarray of <135405x1000 sparse matrix of type '<c
        with 361706 stored elements in Compressed Sparse Row format>>
```

In [77]:
```
X
```

Out[77]:
```
<135405x1000 sparse matrix of type '<class 'numpy.int64'>'
        with 361706 stored elements in Compressed Sparse Row format>
```

In [78]:
```
vocab = vectorizer.vocabulary_
#vocab
```

In [79]:
```
columns=vectorizer.get_feature_names()
#columns
```

```
In [80]: temp7 = DataFrame(X.A, columns=vectorizer.get_feature_names())
         temp7.head(2)
```

Out[80]:

	10	100	101	102	103	104	105	106	107	108	...	wp	wrestling	written	wrong	www	year	years	yes	youtube	zone
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

[2 rows x 1000 columns]

```
In [81]: temp7 = temp7.set_index(wikivand2.index.values)
         temp7.head(2)
```

Out[81]:

	10	100	101	102	103	104	105	106	107	108	...	wp	wrestling	written	wrong	www	year	years	yes	youtube	zone
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

[2 rows x 1000 columns]

```
In [82]: wikivand2 = pd.concat([wikivand2, temp7], axis = 1)
         wikivand2.head(2)
```

Out[82]:

	editid	oldrevisionid	newrevisionid	editcomment	articleid	articletitle	annotators	totalannotators	annotatorid	age	wp	wrestling	written	wrong	www	year	years	yes	youtube	zone
0	1	328391343	328391582	/* Episodes */	24477266	Top Gear (series 14)	3	3	1.0	23.0	0	0	0	0	0	0	0	0	0	0
1	1	328391343	328391582	/* Episodes */	24477266	Top Gear (series 14)	3	3	2.0	23.0	0	0	0	0	0	0	0	0	0	0

[2 rows x 1033 columns]

```
In [83]: # Vectorising ARTICLETITLE
         # creating 500 features from article title
         from sklearn.feature_extraction.text import CountVectorizer
         corpus = wikivand2['articletitle']
         vectorizer = CountVectorizer(min_df = 1, stop_words = 'english', max_featu
         X = vectorizer.fit_transform(corpus)
```

```
X.toarray
#vocab = vectorizer.vocabulary_
temp8 = DataFrame(X.A, columns=vectorizer.get_feature_names())
temp8 = temp8.set_index(wikivand2.index.values)
wikivand2 = pd.concat([wikivand2, temp8], axis = 1)
wikivand2.head(2)
```

Out[83]:

	editid	oldrevisionid	newrevisionid	editcomment	articleid	\
0	1	328391343	328391582	/* Episodes */	24477266	
1	1	328391343	328391582	/* Episodes */	24477266	

	articletitle	annotators	totalannotators	annotatorid	age	\
0	Top Gear (series 14)	3	3	1.0	23.0	
1	Top Gear (series 14)	3	3	2.0	23.0	

	...	williams	winter	wisconsin	womens	world	wrestling	wwe	year
0	...	0	0	0	0	0	0	0	0
1	...	0	0	0	0	0	0	0	0

	york	young
0	0	0
1	0	0

[2 rows x 1533 columns]

In [84]:
```
# Dropping editcommnet and articletitle
wikivand2 = wikivand2.drop( ['editcomment','articletitle'], axis=1)
wikivand2.head(2)
```

Out[84]:

	editid	oldrevisionid	newrevisionid	articleid	annotators	\
0	1	328391343	328391582	24477266	3	
1	1	328391343	328391582	24477266	3	

	totalannotators	annotatorid	age	vandalizing	decisiontime	...	\
0	3	1.0	23.0	0	43453.0	...	
1	3	2.0	23.0	0	14156.0	...	

	williams	winter	wisconsin	womens	world	wrestling	wwe	year	york
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0

	young
0	0
1	0

[2 rows x 1531 columns]

In [85]:
```
# Let's apply Logistic regression to this crazy new dataset
```

```
In [86]: y2 = wikivand2.vandalizing.astype(int)
         x2 = wikivand2.drop(['vandalizing'], axis = 1)

In [87]: # Defining x_train, y_train, x_test, y_test
         from sklearn.cross_validation import train_test_split
         x2_train, x2_test, y2_train, y2_test = train_test_split(x2, y2, test_size=

In [88]: from sklearn.linear_model import LogisticRegression
         from sklearn import datasets
         from sklearn import metrics
         model3 = LogisticRegression(class_weight = 'balanced')
         model3.fit(x2_train,y2_train)

Out[88]: LogisticRegression(C=1.0, class_weight='balanced', dual=False,
                   fit_intercept=True, intercept_scaling=1, max_iter=100,
                   multi_class='ovr', n_jobs=1, penalty='l2', random_state=None,
                   solver='liblinear', tol=0.0001, verbose=0, warm_start=False)

In [89]: # make predictions
         expected = y2_test
         predicted = model3.predict(x2_test)
         # summarize the fit of the model
         print(metrics.classification_report(expected, predicted))

             precision    recall  f1-score   support

          0       0.99      0.72      0.84     26885
          1       0.01      0.47      0.02       196

avg / total       0.99      0.72      0.83     27081



In [90]: print(metrics.confusion_matrix(expected, predicted))

[[19485  7400]
 [  104    92]]


In [91]: model3.score(x2_test, y2_test)

Out[91]: 0.72290535800007383

In [92]: # again an accuray of around 99 %

In [93]: '''
         # Support Vector Machine

         from sklearn import metrics
         from sklearn.svm import SVC
```

```
# fit a SVM model to the data
model4 = SVC()
model4.fit(x2_train, y2_train)

# make predictions
expected4 = y2_test
predicted4 = model2.predict(x2_test)
# summarize the fit of the model
print(metrics.classification_report(expected4, predicted4))
print(metrics.confusion_matrix(expected4, predicted4))
print(model4.score(x2_test, y2_test))
'''
```

Out[93]: '\n# Support Vector Machine\n\nfrom sklearn import metrics\nfrom sklearn.s

In [94]:
```
# Now, I will some visualisation on wikivand1 dataset
# it has all the columns as int or float except article name or comments
# I want to get an idea about how age, time-taken in edit affect vandalism
```

In [95]:
```
wikivand1.head()
```

Out[95]:

	editid	oldrevisionid	newrevisionid	articleid	annotators	\
0	1	328391343	328391582	24477266	3	
1	1	328391343	328391582	24477266	3	
2	1	328391343	328391582	24477266	3	
3	2	327585467	327607921	476288	10	
4	2	327585467	327607921	476288	10	

	totalannotators	annotatorid	age	vandalizing	decisiontime	\
0	3	1.0	23.0	0	43453.0	
1	3	2.0	23.0	0	14156.0	
2	3	3.0	26.0	1	22190.0	
3	18	1.0	23.0	0	24203.0	
4	18	2.0	23.0	0	25283.0	

	...	editing_never	editing_less	editing_monthly	\
0	...	0	1	0	
1	...	0	1	0	
2	...	0	1	0	
3	...	0	1	0	
4	...	0	1	0	

	editing_weekly	editing_daily	noticing_less	noticing_monthly	\
0	0	0	0	0	
1	0	0	0	1	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	1	

```
   noticing_never  noticing_weekly  noticing_daily
0               0                0               1
1               0                0               0
2               1                0               0
3               0                0               1
4               0                0               0

[5 rows x 31 columns]
```

```
In [96]: wikivand1.columns
```

```
Out[96]: Index(['editid', 'oldrevisionid', 'newrevisionid', 'articleid', 'annotator
               'totalannotators', 'annotatorid', 'age', 'vandalizing', 'decisionti
               'time', 'class_y_no', 'class_y_yes', 'class_y_dunno', 'class_y_erro
               'class_x_lb', 'sex_lb', 'reading_daily', 'reading_weekly',
               'reading_monthly', 'reading_less', 'editing_never', 'editing_less',
               'editing_monthly', 'editing_weekly', 'editing_daily', 'noticing_les
               'noticing_monthly', 'noticing_never', 'noticing_weekly',
               'noticing_daily'],
              dtype='object')
```

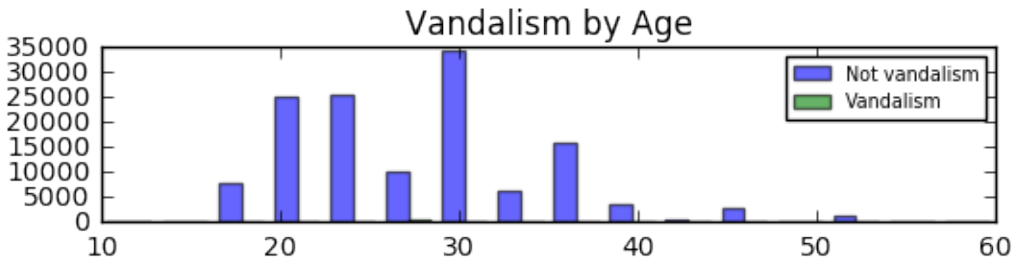
```
In [97]: import matplotlib.pyplot as plt
         alpha = 0.6
         fig = plt.figure(figsize=(8, 12))
         grouped = wikivand1.groupby(['vandalizing'])
         group0 = grouped.get_group(0)
         group1 = grouped.get_group(1)
```

```
<matplotlib.figure.Figure at 0x121bc0048>
```

```
In [98]: plot_rows = 3
         plot_cols = 1
```

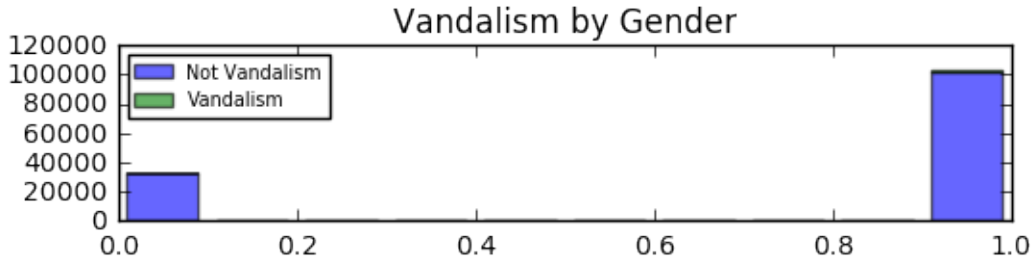
```
In [99]: #ax1 = fig.add_subplot(2,2,1)
         ax1 = plt.subplot2grid((plot_rows,plot_cols), (0,0), rowspan=1, colspan=1)
         plt.hist([group0.age, group1.age], bins=16, range=(10,60), stacked=False,
                  label=['Not vandalism', 'Vandalism'], alpha=alpha)
         plt.legend(loc='best', fontsize='x-small')
         ax1.set_title('Vandalism by Age')
```

```
Out[99]: <matplotlib.text.Text at 0x123326160>
```



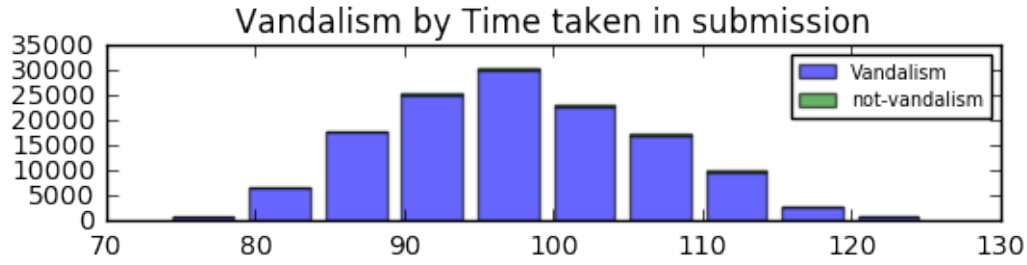
```
In [100]: ax2 = plt.subplot2grid((plot_rows,plot_cols), (0,0), rowspan=1, colspan=1
          plt.hist([group0.sex_lb, group1.sex_lb],  stacked=True,
                   label=['Not Vandalism', 'Vandalism'], alpha=alpha)
          plt.legend(loc='best', fontsize='x-small')
          ax2.set_title('Vandalism by Gender')

Out[100]: <matplotlib.text.Text at 0x1233b69b0>
```




```
In [101]: ax3 = plt.subplot2grid((plot_rows,plot_cols), (0,0), rowspan=1, colspan=1
          plt.hist([group0.time, group1.time],  stacked=True,
                   label=['Vandalism', 'not-vandalism'], alpha=alpha)
          plt.legend(loc='best', fontsize='x-small')
          ax3.set_title('Vandalism by Time taken in submission')

Out[101]: <matplotlib.text.Text at 0x126e9ecf8>
```




```
In [102]: wikivand2['vandalizing'].value_counts()

Out[102]: 0    134413
          1       992
          Name: vandalizing, dtype: int64

In [103]: wikivand2['sex_lb'].value_counts()
```

```
Out[103]: 1    103069
          0     32336
          Name: sex_lb, dtype: int64
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```