Business Problem:

Given a user comment, we need to identify the classes of toxicity it will fall into. This is a Multi-label classification (toxic, severe_toxic, obscene, threat, insult, identity_hate)

Exploratory Data Analysis:

```
df.head()
```

	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	0000997932d777bf	Explanation\nWhy the edits made under my username Hardcore Metallica Fan were reverted? They weren't vandalisms, just closure on some GAs after I voted at New York Dolls FAC. And please don't remove the template from the talk page since I'm retired now.89.205.38.27	0	0	0	0	0	0
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm seemingly stuck with. Thanks. (talk) 21:51, January 11, 2016 (UTC)	0	0	0	0	0	0
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It's just that this guy is constantly removing relevant information and talking to me through edits instead of my talk page. He seems to care more about the formatting than the actual info.	0	0	0	0	0	0
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on improvement - I wondered if the section statistics should be later on, or a subsection of ""types of accidents"" -I think the references may need tidying so that they are all in the exact same format ie date format etc. I can do that later on, if no-one else does first - if you have any preferences for formatting style on references or want to do it yourself please let me know.\n\nThere appears to be a backlog on articles for review so I guess there may be a delay until a reviewer turns up. It's listed in the relevant form eg Wikipedia:Good_article_nominations#Transport "	0	0	0	0	0	0
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember what page that's on?	0	0	0	0	0	0

```
df.info()
```

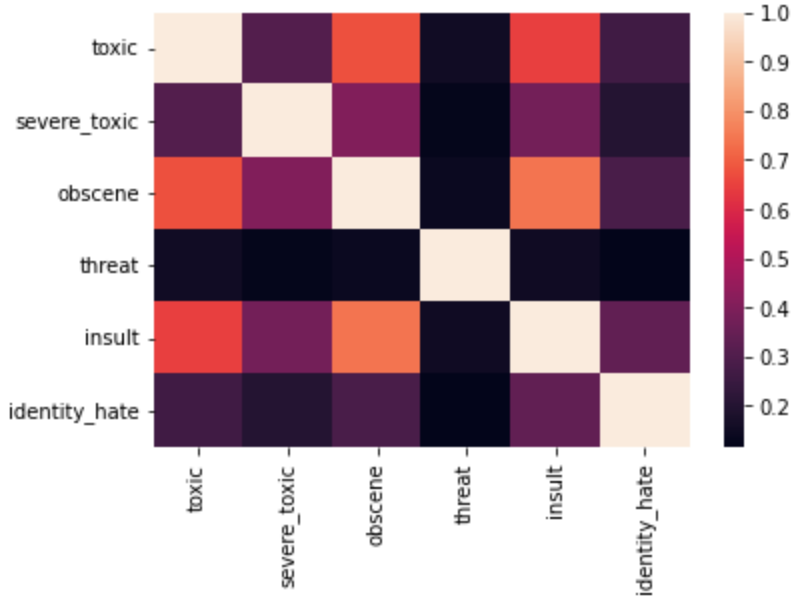
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 159571 entries, 0 to 159570
Data columns (total 8 columns):
 #   Column         Non-Null Count   Dtype 
---  ------         --------------   ----- 
 0   id             159571 non-null  object
 1   comment_text   159449 non-null  object
 2   toxic          159571 non-null  int64 
 3   severe_toxic   159571 non-null  int64 
 4   obscene        159571 non-null  int64 
 5   threat         159571 non-null  int64 
 6   insult         159571 non-null  int64 
 7   identity_hate  159571 non-null  int64 
dtypes: int64(6), object(2)
memory usage: 9.7+ MB
```

```
df.describe()
```

	toxic	severe_toxic	obscene	threat	insult	identity_hate
count	159571.000000	159571.000000	159571.000000	159571.000000	159571.000000	159571.000000
mean	0.095844	0.009996	0.052948	0.002996	0.049364	0.008805
std	0.294379	0.099477	0.223931	0.054650	0.216627	0.093420
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

```
sns.heatmap(df.corr())
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f14916cfe50>
```



identity_hate

Data preprocessing:

- Removing special characters.
- Converting everything to lowercase
- Stop word removal
- Lemmatization
- Removing extra space characters
- Removing words with less than 2 characters
- Removing NULL values

Vectorization:

Tf-Idf weighted word2vector

```
print("Shape of X-Train data=",X_train_tfidf_w2v.shape)
print("Shape of X-val data=",X_val_tfidf_w2v.shape)
print("Shape of X-Test data=",X_test_tfidf_w2v.shape)
```

```
Shape of X-Train data= (111614, 300)
Shape of X-val data= (79724, 300)
Shape of X-Test data= (79725, 300)
```

```
print("Shape of y-Train data=",y_train.shape)
print("Shape of y-val data=",y_val.shape)
print("Shape of y-Test data=",y_test.shape)
```

```
Shape of y-Train data= (111614, 6)
Shape of y-val data= (79724, 6)
Shape of y-Test data= (79725, 6)
```

Baseline Models:

BinaryRelevance

```
Accuracy Score : 0.9056130448416432

Average AUC Score : 0.9597570351892837

Hamming loss : 0.023898818856485836

Log loss : 1.487558361966342
```

Label PowerSet

```
Accuracy Score : 0.9099027908435247

Average AUC Score : 0.9632148138876427

Hamming loss : 0.02423748301452911

Log loss : 1.0896428553585606
```

Neural Networks:

DNN:

- accuracy: 0.9670
- avg_AUC: 0.9728
- Hamming_Loss: 0.0205
- Log_Loss: 1.6014

Bi-directional LSTM:

- accuracy: 0.9917
- avg_AUC: 0.5366
- Hamming_Loss: 0.0432
- Log_Loss: 0.4641

Confusion Matrix:

DNN:

```
[[[70480  1603]
  [ 2389  5253]]

 [[78909    24]
  [  758    34]]

 [[75103   403]
  [ 1664  2555]]

 [[79482     4]
  [  227    12]]

 [[74741  1050]
  [ 1482  2452]]

 [[78929    98]
  [  551   147]]]
```

	precision	recall	f1-score	support
0	0.77	0.69	0.72	7642
1	0.59	0.04	0.08	792
2	0.86	0.61	0.71	4219
3	0.75	0.05	0.09	239
4	0.70	0.62	0.66	3934
5	0.60	0.21	0.31	698
micro avg	0.77	0.60	0.67	17524
macro avg	0.71	0.37	0.43	17524
weighted avg	0.76	0.60	0.65	17524
samples avg	0.06	0.05	0.05	17524

Bi-directional LSTM:

```
[[[68701  3382]
  [ 7482   160]]

 [[78933     0]
  [  792     0]]

 [[75506     0]
  [ 4219     0]]

 [[79486     0]
  [  239     0]]

 [[75791     0]
  [ 3934     0]]

 [[79027     0]
  [  698     0]]]
              precision    recall  f1-score   support

           0       0.05      0.02      0.03      7642
           1       0.00      0.00      0.00       792
           2       0.00      0.00      0.00      4219
           3       0.00      0.00      0.00       239
           4       0.00      0.00      0.00      3934
           5       0.00      0.00      0.00       698

   micro avg       0.05      0.01      0.02     17524
   macro avg       0.01      0.00      0.00     17524
weighted avg       0.02      0.01      0.01     17524
 samples avg       0.00      0.00      0.00     17524
```

Future work:

- Splitting the train, test and validation datasets using stratification technique.
- Performing Hyper-parameter tuning for the Deep Learning models using Optuna library.
- Using Bert
- …