## 数据分析

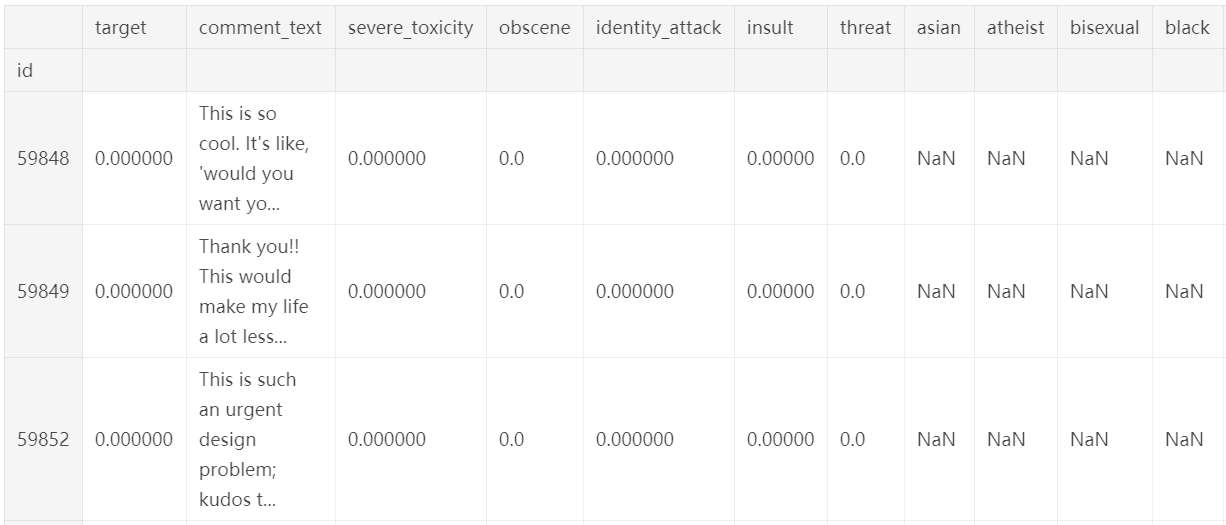
数据集共两个文件train.csv和test.csv两个文件。test.csv只包含文本和id。train.csv在此基础上还包含对应的属性值。comment\_text列为评论内容，target列为对应的标签（即toxicity label），模型的任务就是预测test.csv文件的target值，target>=0.5的为恶意评论（toxic）。

训练集文件还包含有一些恶意评论的子属性：severe\_toxicity,obscene,threat,

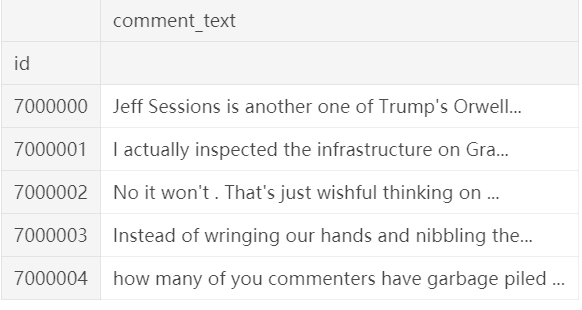
insult, identity\_attack, sexual\_explicit和评论的其他相关信息列：asian, male，female，transgender, other\_gender, black,white, Christian, jewish, muslim等。

train\_df = pd.read\_csv(os.path.join(JIGSAW\_PATH,'train.csv'), index\_col='id')    
test\_df = pd.read\_csv(os.path.join(JIGSAW\_PATH,'test.csv'), index\_col='id')

（1）训练集（train.csv）数据预览

train.head()

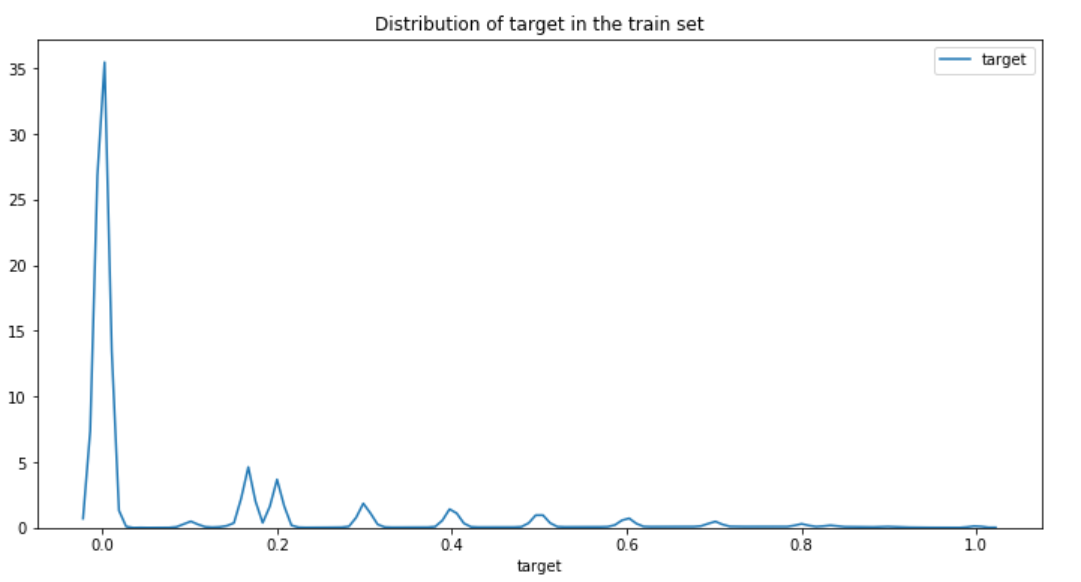
（2）测试集（test.csv）数据预览

test\_df.head()

### 1.1 数据分布

通过绘制分布直方图，了解数据的基本分布特征。

#### （1）target的分布

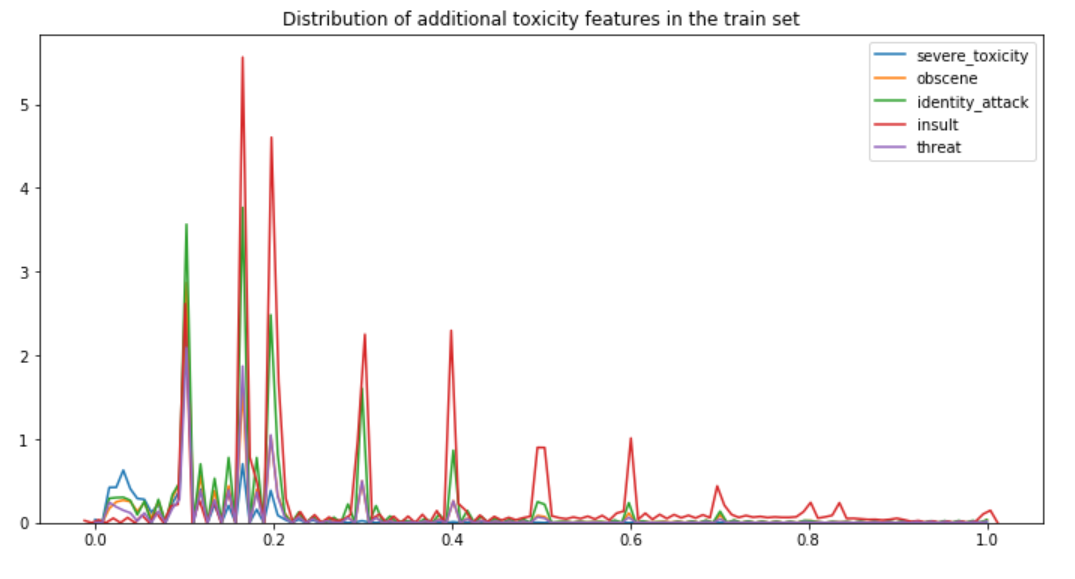
plt.figure(figsize=(12,6))  
plt.title("Distribution of target in the train set")    
sns.distplot(train['target'],kde=True,hist=False, bins=120, label='target')    
plt.legend();   
plt.show()

#### （2）toxic feature的分布

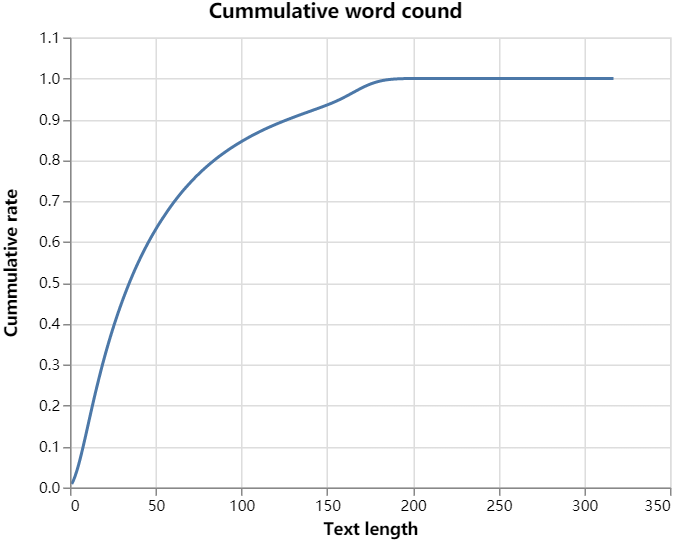
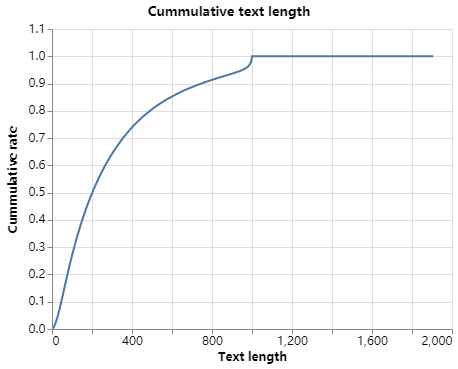
恶意评论相关子类型的分布；

**def** plot\_features\_distribution(features, title):      
  plt.figure(figsize=(12,6))    
    plt.title(title)    
    **for** feature **in** features:    
        sns.distplot(train\_df[feature],kde=True,hist=False,bins=120, label=feature)    plt.xlabel('')    
    plt.legend()    
    plt.show()

features = ['severe\_toxicity', 'obscene','identity\_attack','insult','threat']  
plot\_features\_distribution(features, "Distribution of additional toxicity features in the train set")



#### （3）文本长度分布

text\_length = train['comment\_text'].apply(**lambda** x: len(x)).value\_counts(normalize=True).sort\_index().cumsum().reset\_index().rename(columns={'index': 'Text length'})

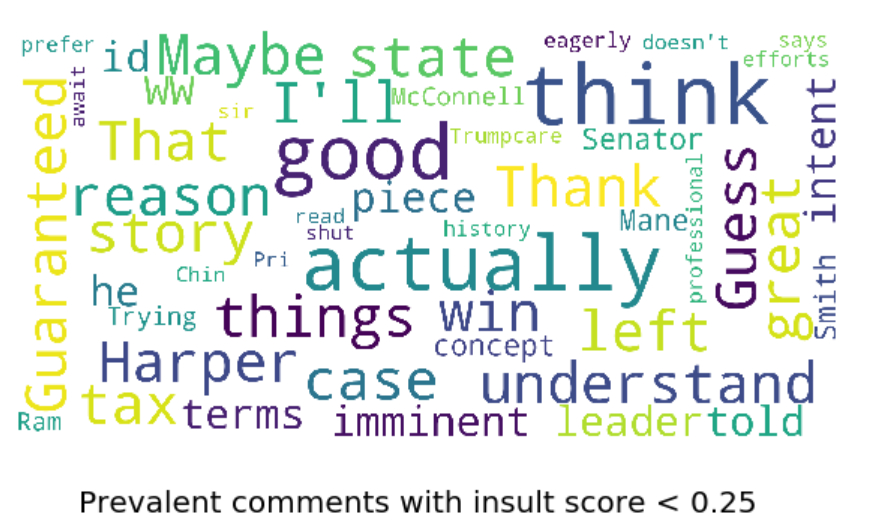
长度为1000的文本数量较多，1000应该是默认的最大长度。95%的文本长度小于125词。

### 高频词云

绘制高频词云图，统计各个类型数据的高频词，特征词。

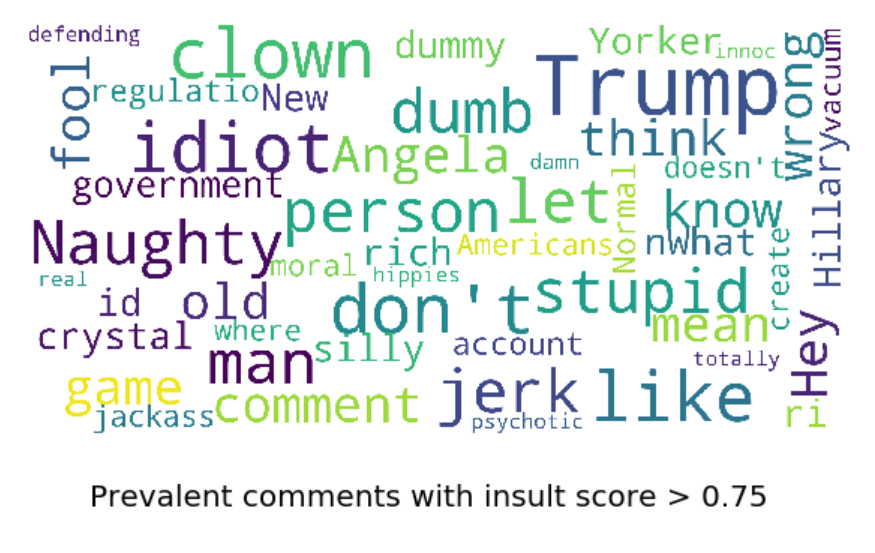
#### （1） insult

Insult score 小于0.25的评论高频词云。多半为正面词汇。

show\_wordcloud(train.loc[train['insult'] < 0.25]['comment\_text'].sample(20000), title = 'Prevalent comments with insult score < 0.25')

Insult score 大于0.75的评论高频词云，多半为负面词汇和特定人名等

show\_wordcloud(train.loc[train['insult'] > 0.75]['comment\_text'].sample(20000), title = 'Prevalent comments with insult score > 0.75')



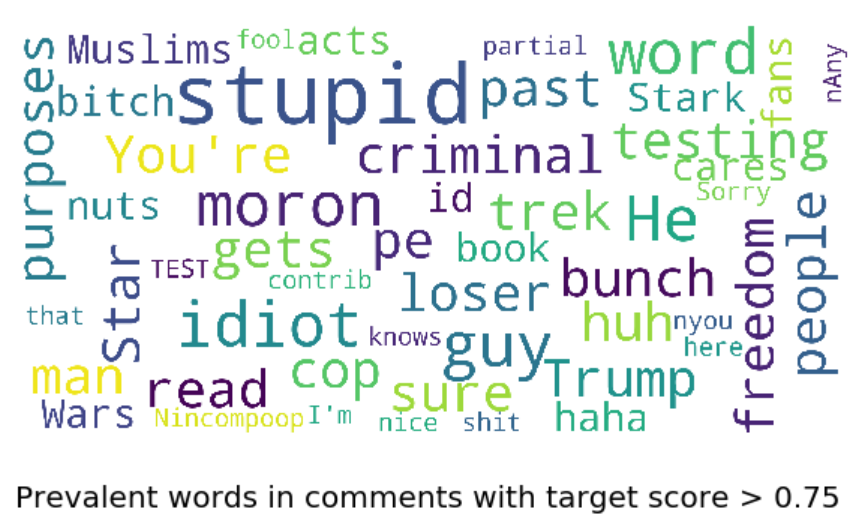
#### （2）target高频词云

target score<0.25高频词云。多半为正面词汇。

show\_wordcloud(train.loc[train['target'] < 0.25]['comment\_text'],title = 'Prevalent words in comments with target score < 0.25')

target score>0.75高频词云。多半为负面词汇。

show\_wordcloud(train.loc[train['target'] > 0.75]['comment\_text'],title = 'Prevalent words in comments with target score > 0.75')



可以看出，是否为toxic恶意评论和其中的某些特定关键词有较大联系。

## 数据预处理

在将数据送入模型之前，对数据集中的一些错误和异常数据进行简单处理。

（1）缩写恢复

将常见缩写形式恢复，便于对单词编码和后期处理。

contraction\_mapping = {"ain't": "is not", "aren't": "are not", "can't": "cannot", "'cause": "because","could've": "could have","demonitization": "demonetization", "demonetisation": "demonetization", …}

（2）常见错写恢复

一些常用表达感情的词汇，在很多文本中会故意复写或漏写、错写某些单词，表达原词含义或加深，在这里对齐进行恢复，以便统一处理。

RE\_PATTERNS = {' american ': ['amerikan'],    
' fuck ': ['f@ck', 'f\-ing', 'fux', 'feck ', 'f\\*\\*',  'f\- ing', 'f\.u\.', 'f###', ' fu ', 'f@ck', 'f u c k', 'f uck', 'f ck'],  ' bitch ': ['b[w]\*i[t]\*ch', 'b!tch',  'bi\+ch', 'b!\+ch', '(b)([^a-z]\*)(i)([^a-z]\*)(t)([^a-z]\*)(c)([^a-z]\*)(h)',   
 'biatch', 'bi\\*\\*h', 'bytch', 'b i t c h'],   
 ' shit ': ['shitty', '(s)([^a-z ]\*)(h)([^a-z ]\*)(i)([^a-z ]\*)(t)', 'shite', '\$hit', 's h i t'],  ' sex ': ['sexy', 's3x', 'sexuality']}