

```
In [2]: import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn import model_selection, naive_bayes, svm
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, f1_score
from tqdm import tqdm
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MaxAbsScaler
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import roc_auc_score
import os.path
import pickle
```

Reading test and train data from already preprocessed pickle file

```
In [4]: X_train = pd.read_pickle('../.../Preprocessing/Data/X_train.pkl')
X_test = pd.read_pickle('../.../Preprocessing/Data/X_test.pkl')
y_train = pd.read_pickle('../.../Preprocessing/Data/y_train.pkl')
y_test = pd.read_pickle('../.../Preprocessing/Data/y_test.pkl')
```

Performing TF-IDF over the dataset for word embedding

```
In [5]: vec = TfidfVectorizer(ngram_range=(1, 2), min_df=3,
                             max_df=0.9, strip_accents='unicode', use_idf=1, smooth_idf=1, sublinear_tf=1)
df = pd.read_pickle('../.../Preprocessing/Data/preprocess_pickle.pkl')
df['comment_text_final'] = [" ".join(text) for text in df['comment_text_final'].values]
vec = vec.fit(df['comment_text_final'])
```

```
In [6]: train_term_doc = vec.transform(X_train)
test_term_doc = vec.transform(X_test)
```

Scaling the input data using MaxAbsScaler

```
In [7]: scaler = MaxAbsScaler()
train_term_doc = scaler.fit_transform(train_term_doc)
test_term_doc = scaler.fit_transform(test_term_doc)
```

Performing Cross Validation for multiple values of C and also using Ridge and Lasso Regularization for Logistic Regression

```

In [12]: def logistic_regression_with_CV(label):
            if os.path.isfile('Models/ridge_lr_' + label + '.sav') and os.path.isfile('Models/lasso_lr_' + label + '.sav'):
                ridge_logistic_regressor_grid_cv = pickle.load(open('Models/ridge_lr_' + label + '.sav', 'rb'))
                lasso_logistic_regressor_grid_cv = pickle.load(open('Models/lasso_lr_' + label + '.sav', 'rb'))
            else:
                ridge_logistic_regressor = LogisticRegression(penalty="l2", solver="liblinear", max_iter = 2000)
                lasso_logistic_regressor = LogisticRegression(penalty="l1", solver="liblinear", max_iter = 2000)

                ridge_logistic_regressor_grid_cv = GridSearchCV(estimator=ridge_logistic_regressor,
                                                                param_grid={'C': np.logspace(-4, 4, 20)}, cv= 5, iid=False, n_jobs=-1)
                lasso_logistic_regressor_grid_cv = GridSearchCV(estimator=lasso_logistic_regressor,
                                                                param_grid={'C': np.logspace(-4, 4, 20)}, cv= 5, iid=False, n_jobs=-1)

                ridge_logistic_regressor_grid_cv.fit(train_term_doc, y_train[label])
                lasso_logistic_regressor_grid_cv.fit(train_term_doc, y_train[label])

                pickle.dump(ridge_logistic_regressor_grid_cv, open('Models/ridge_lr_' + label + '.sav', 'wb'))
                pickle.dump(lasso_logistic_regressor_grid_cv, open('Models/lasso_lr_' + label + '.sav', 'wb'))

                ridge_train_pred = ridge_logistic_regressor_grid_cv.predict(train_term_doc)
                lasso_train_pred = lasso_logistic_regressor_grid_cv.predict(train_term_doc)

                ridge_test_pred = ridge_logistic_regressor_grid_cv.predict(test_term_doc)
                lasso_test_pred = lasso_logistic_regressor_grid_cv.predict(test_term_doc)

                print(label + " Ridge Train Accuracy - " + str(ridge_logistic_regressor_grid_cv.score(train_term_doc, y_train[label])))
                print(label + " Lasso Train Accuracy - " + str(lasso_logistic_regressor_grid_cv.score(train_term_doc, y_train[label])) + '\n')

                print(label + " Ridge Train F1 Score - " + str(f1_score(y_train[label], ridge_train_pred)))
                print(label + " Lasso Train F1 Score - " + str(f1_score(y_train[label], lasso_train_pred)) + '\n')

                print(label + " Ridge Train ROC-AUC Score - " + str(roc_auc_score(y_train[label], ridge_train_pred)))
                print(label + " Lasso Train ROC-AUC Score - " + str(roc_auc_score(y_train[label], lasso_train_pred)) + '\n')

```

```
print(label + " Ridge Test Accuracy - " + str(ridge_logistic_regress
or_grid_cv.score(test_term_doc, y_test[label])))
print(label + " Lasso Test Accuracy - " + str(lasso_logistic_regress
or_grid_cv.score(test_term_doc, y_test[label])) + '\n')

print(label + " Ridge Test F1 Score - " + str(f1_score(y_test[label
], ridge_test_pred)))
print(label + " Lasso Test F1 Score - " + str(f1_score(y_test[label
], lasso_test_pred)) + '\n')

print(label + " Ridge Test ROC-AUC Score - " + str(roc_auc_score(y_t
est[label], ridge_test_pred)))
print(label + " Lasso Test ROC-AUC Score - " + str(roc_auc_score(y_t
est[label], lasso_test_pred)) + '\n\n')
```

Accuracy, F1 and ROC-AUC score for each label using Binary Relevance Technique

```
In [14]: label_cols = ['toxic', 'severe_toxic', 'obscene', 'threat', 'insult', 'i  
         identity_hate']  
         for label in label_cols:  
             logistic_regression_with_CV(label)
```

```
toxic Ridge Train Accuracy - 0.9980958107837242
toxic Lasso Train Accuracy - 0.9608889557025456

toxic Ridge Train F1 Score - 0.9963052989790958
toxic Lasso Train F1 Score - 0.9214828227956341

toxic Ridge Train ROC-AUC Score - 0.9970702556227102
toxic Lasso Train ROC-AUC Score - 0.9375781202131243

toxic Ridge Test Accuracy - 0.8998423114095325
toxic Lasso Test Accuracy - 0.9124573986469302

toxic Ridge Test F1 Score - 0.7907322776065469
toxic Lasso Test F1 Score - 0.8218610909843701

toxic Ridge Test ROC-AUC Score - 0.8487089790281281
toxic Lasso Test ROC-AUC Score - 0.8736658098360226


severe_toxic Ridge Train Accuracy - 0.9798306273802365
severe_toxic Lasso Train Accuracy - 0.9747945480056124

severe_toxic Ridge Train F1 Score - 0.4459738472126635
severe_toxic Lasso Train F1 Score - 0.3119015047879617

severe_toxic Ridge Train ROC-AUC Score - 0.6514190451932962
severe_toxic Lasso Train ROC-AUC Score - 0.604954842315839

severe_toxic Ridge Test Accuracy - 0.9750241619614426
severe_toxic Lasso Test Accuracy - 0.97319293962053

severe_toxic Ridge Test F1 Score - 0.2832116788321168
severe_toxic Lasso Test F1 Score - 0.3111111111111111

severe_toxic Ridge Test ROC-AUC Score - 0.5898471374528302
severe_toxic Lasso Test ROC-AUC Score - 0.609046666288601


obscene Ridge Train Accuracy - 0.9977951493285228
obscene Lasso Train Accuracy - 0.9683553818400481

obscene Ridge Train F1 Score - 0.9922054915854739
obscene Lasso Train F1 Score - 0.8816640119928792

obscene Ridge Train ROC-AUC Score - 0.9948022231187991
obscene Lasso Train ROC-AUC Score - 0.9114942423937477

obscene Ridge Test Accuracy - 0.935958085355308
obscene Lasso Test Accuracy - 0.9445546569001475

obscene Ridge Test F1 Score - 0.7470363672895318
obscene Lasso Test F1 Score - 0.7930117736422331

obscene Ridge Test ROC-AUC Score - 0.822981526631658
obscene Lasso Test ROC-AUC Score - 0.8621685999837063
```

```
threat Ridge Train Accuracy - 0.9998747243936661
threat Lasso Train Accuracy - 0.9997995590298657

threat Ridge Train F1 Score - 0.9925925925925926
threat Lasso Train F1 Score - 0.9882005899705014

threat Ridge Train ROC-AUC Score - 0.9926470588235294
threat Lasso Train ROC-AUC Score - 0.9926091532337185

threat Ridge Test Accuracy - 0.9930311816470827
threat Lasso Test Accuracy - 0.9914034284551605

threat Ridge Test F1 Score - 0.3800904977375566
threat Lasso Test F1 Score - 0.3157894736842105

threat Ridge Test ROC-AUC Score - 0.6511237619241709
threat Lasso Test ROC-AUC Score - 0.6395114068906841


insult Ridge Train Accuracy - 0.9843154940869914
insult Lasso Train Accuracy - 0.9283423531769894

insult Ridge Train F1 Score - 0.9388910581803983
insult Lasso Train F1 Score - 0.6805896805896805

insult Ridge Train ROC-AUC Score - 0.9549673465724602
insult Lasso Train ROC-AUC Score - 0.7805078981576883

insult Ridge Test Accuracy - 0.9191718805636095
insult Lasso Test Accuracy - 0.9243094765756142

insult Ridge Test F1 Score - 0.646496106785317
insult Lasso Test F1 Score - 0.686472819216182

insult Ridge Test ROC-AUC Score - 0.7647524522277466
insult Lasso Test ROC-AUC Score - 0.7961135267715213


identity_hate Ridge Train Accuracy - 0.9970685508117859
identity_hate Lasso Train Accuracy - 0.9799809581078373

identity_hate Ridge Train F1 Score - 0.9365853658536586
identity_hate Lasso Train F1 Score - 0.3923954372623574

identity_hate Ridge Train ROC-AUC Score - 0.9511026105598092
identity_hate Lasso Train ROC-AUC Score - 0.6335127059636773

identity_hate Ridge Test Accuracy - 0.9788392085050104
identity_hate Lasso Test Accuracy - 0.9798565542499619

identity_hate Ridge Test F1 Score - 0.33118971061093244
identity_hate Lasso Test F1 Score - 0.38317757009345804

identity_hate Ridge Test ROC-AUC Score - 0.6131074575020263
identity_hate Lasso Test ROC-AUC Score - 0.6354288860734549
```

In []: