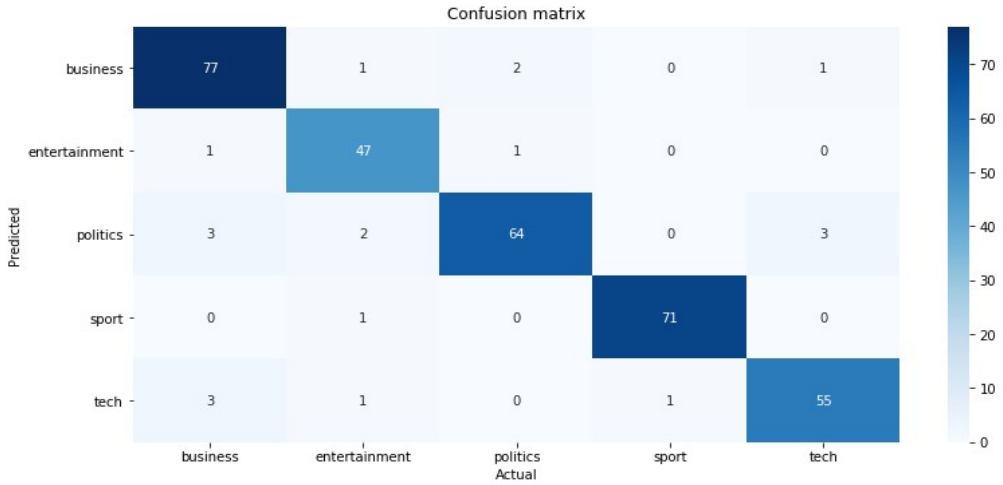
Fatahillah Karomy
1301164346

TEXT CLASSIFICATION

Pada program yang sudah diberikan akurasi proses klasifikasi menghasilkan 94% begitu juga untuk base modelnya

```
In [125]: base_model = LogisticRegression(random_state = 8)
          base_model.fit(features_train, labels_train)
          accuracy_score(labels_test, base_model.predict(features_test))
Out[125]: 0.9401197604790419

In [126]: best_classifier.fit(features_train, labels_train)
          accuracy_score(labels_test, best_classifier.predict(features_test))
Out[126]: 0.9401197604790419
```

dan berikut juga confusion matrix pada program ini



A. Hasil Klasifikasi dengan menghilangkann proses normalisasi

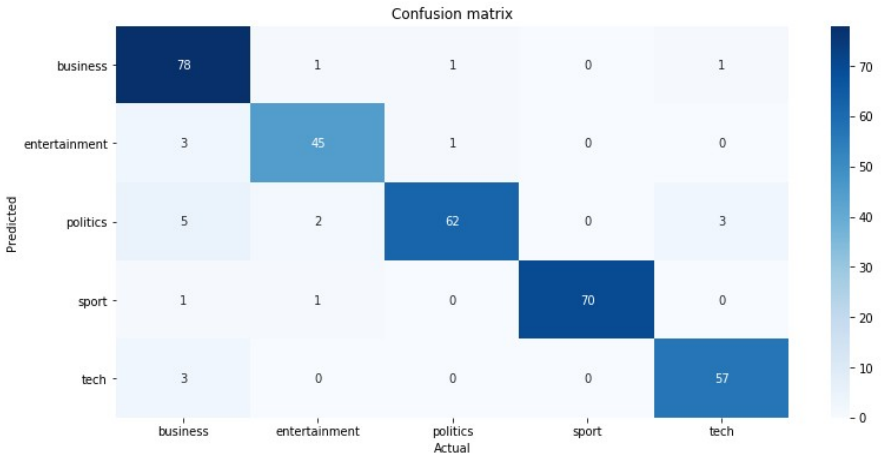
pada bagian ini saya menambahkann cell baru pada program dimana pada dasarnya sama tapi tidak memasukkan bagian normalisasi, proses normalisasi dimana proses tersebut menghilangkan beberapa huruf seperti ‘s atau simbol simbol lainnya mengubah, semua huruf menjadi lower case atau upper case. Lalu menyimpanya dalam bentuk file _withoutnormalization untuk setiap file yang digunakan

```
In [212]: # X_train
with open('Data/X_train_withoutnormalization.pickle', 'wb') as output:
    pickle.dump(X_train, output)

# X_test
with open('Data/X_test_withoutnormalization.pickle', 'wb') as output:
    pickle.dump(X_test, output)

# y_train
with open('Data/y_train_withoutnormalization.pickle', 'wb') as output:
    pickle.dump(y_train, output)

# y_test
with open('Data/y_test_withoutnormalization.pickle', 'wb') as output:
    pickle.dump(y_test, output)

# df
with open('Data/df_withoutnormalization.pickle', 'wb') as output:
    pickle.dump(df, output)

# features_train
with open('Data/features_train_withoutnormalization.pickle', 'wb') as output:
    pickle.dump(features_train, output)

# labels_train
with open('Data/labels_train_withoutnormalization.pickle', 'wb') as output:
    pickle.dump(labels_train, output)

# features_test
with open('Data/features_test_withoutnormalization.pickle', 'wb') as output:
    pickle.dump(features_test, output)

# labels_test
with open('Data/labels_test_withoutnormalization.pickle', 'wb') as output:
    pickle.dump(labels_test, output)

# TF-IDF object
with open('Data/tfidf_withoutnormalization.pickle', 'wb') as output:
    pickle.dump(tfidf, output)
```

hasil dari program tersebut adalah ketika proses normalisasi dihilangkan

```
In [144]: base_model = LogisticRegression(random_state = 8)
          base_model.fit(features_train, labels_train)
          accuracy_score(labels_test, base_model.predict(features_test))
Out[144]: 0.9281437125748503

In [146]: best_classifier.fit(features_train, labels_train)
          accuracy_score(labels_test, best_classifier.predict(features_test))
Out[146]: 0.9341317365269461
```

dimana base model nya 92,8 % dan best classifier 93,4 % dimana akurasi lebih kecil dibanding dengan adanya normalisasi karena salah satu penyebabnya adalah huruf uppercase dan lowercase merupakan sebuah huruf yang berbeda yang mana menyebabkan akurasi menurun
dan berikut juga confusion matrix pada bagian proses ini



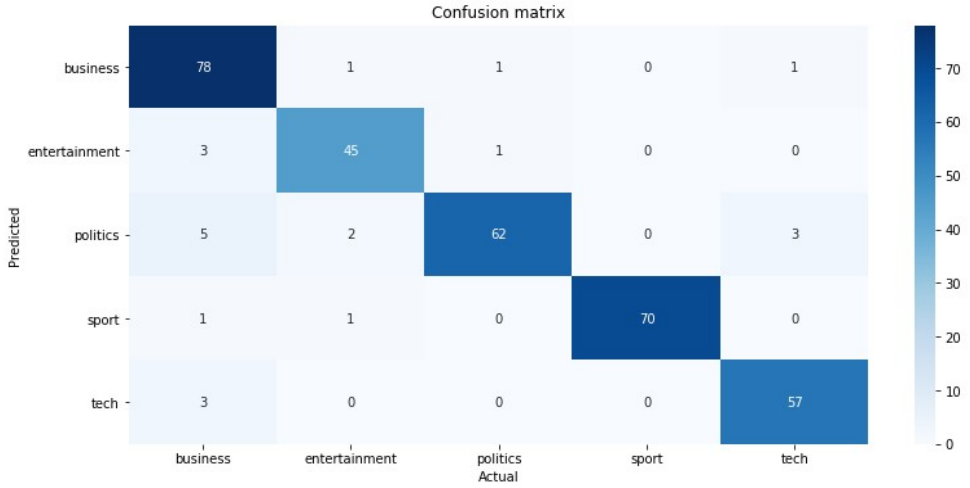
B. Hasil Klasifikasi tanpa lemmatisasi

lemmatisasi adalah proses dimana mengubah kata menjadi kata dasar contohnya computer menjadi compute disini melalkukan hal sama dimana membuat file baru _withoutlemmtization diamanan untuk menyimpan extraksi tanpa lemmatization
hasil akurasi tanpa lemmatisasi

```
In [160]: base_model = LogisticRegression(random_state = 8)
          base_model.fit(features_train, labels_train)
          accuracy_score(labels_test, base_model.predict(features_test))
Out[160]: 0.9401197604790419

In [161]: best_classifier.fit(features_train, labels_train)
          accuracy_score(labels_test, best_classifier.predict(features_test))
Out[161]: 0.9461077844311377
```

disini hasil akurasi untuk base model 94% dan best classifier 94,6%

confusion matrix dari menghilangkan proses lemmatisasi

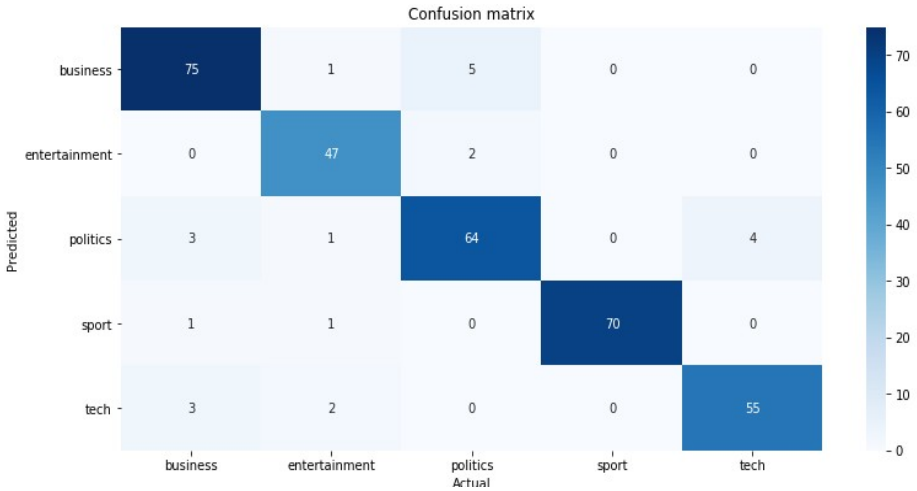


C. Tanpa menggunakan stop word

stop word merupakan kata kata yang pada dasarnya tidak terlalu penting atau yang tidak memiliki makna sama sekali seperti “the”, “a”, “on”, “is”, “all” hasil dari tanpa menghapus stop word mendapatkan akurasi

```
In [196]: base_model = LogisticRegression(random_state = 8)
          base_model.fit(features_train, labels_train)
          accuracy_score(labels_test, base_model.predict(features_test))
Out[196]: 0.9221556886227545

In [197]: best_classifier.fit(features_train, labels_train)
          accuracy_score(labels_test, best_classifier.predict(features_test))
Out[197]: 0.9311377245508982
```

base model 92,2% dan best classifier sebesar 93,1%
untuk confusion matrix



Hasil mengurangi max features pada tf idf

disini mencoba mengurangi max features, dimana memberi nilai max features menjadi 100, lalu hasil akurasi dari max features 100 adalah

```
In [226]: base_model = LogisticRegression(random_state = 8)
          base_model.fit(features_train, labels_train)
          accuracy_score(labels_test, base_model.predict(features_test))
Out[226]: 0.8802395209580839

In [227]: best_classifier.fit(features_train, labels_train)
          accuracy_score(labels_test, best_classifier.predict(features_test))
Out[227]: 0.8922155688622755
```

dimana base model 88% dan best classifier 89% yang mana menurun dibandingkan sebelumnya yang memiliki max features 300

Hasil menambah max features pada tf idf

disini mencoba mengurangi max features, dimana memberi nilai max features menjadi 600, lalu hasil akurasi dari max features 600 adalah

```
In [244]: base_model = LogisticRegression(random_state = 8)
          base_model.fit(features_train, labels_train)
          accuracy_score(labels_test, base_model.predict(features_test))
Out[244]: 0.9610778443113772

In [245]: best_classifier.fit(features_train, labels_train)
          accuracy_score(labels_test, best_classifier.predict(features_test))
Out[245]: 0.9670658682634731
```

dimana base model 96,1% dan best classifier 96,7% yang mana meningkat dibandingkan sebelumnya yang memiliki max features 300

Random Forest Classifier

Disini menggunakan Random Forest Classifier tetapi tidak bisa dilanjutkan terdapat error saat mencari best parameter tetepi mean accuracy yang didapatkan sebesar 93,9%

```
The mean accuracy of a model with these hyperparameters is:
0.9397160801281245
```

SVM Classifier

Disini menggunakan SVM Classifier tetapi tidak bisa dilanjutkan terdapat error saat mencari best parameter tetapi mean accuracy yang didapatkan sebesar 23,2%

```
The mean accuracy of a model with these hyperparameters is:
0.23215270964874768
```

yang mana disini saya belum menemukan parameter yang optimal untuk svm

Klasifikasi Bahasa Indonesia

Jika anda ingin menggunakan teks bahasa Indonesia, maka dataset yang mana pada bagian **Dataset Creation** lalu pada stopword harus disesuaikan dengan bahasa indonesia