Laporan Tugas Programming NLP: Text Classification

Nama : Muhammad Mukhtar Dwi Putra

NIM : 1301170278

**2. Coba buatkan feature berikut (save dan upload feature), lalu laporkan pengaruhnya terhadap akurasi klasifikasi: a. Tanpa proses normalisation b. Tanpa proses lemmatisation c. Tanpa menghilangkan stopwords**

Jawab :

Dalam proses pencarian akurasi yang dihasilkan untuk menganalisis pemilihan fitur dalam preprocessing, pada setiap 'proses yang dihilangkan', dicari parameter model terbaiknya menggunakan randomize search lalu grid search. Sehingga hasil akurasi yang dihasilkan model sudah memiliki parameter terbaik dan yang dianalisis tinggal akurasi dari 'proses yang dihilangkan' pada preprocessing. Model machine learning yang digunakan adalah logistic regression.

(a) Tanpa proses normalisasi

- Data set tidak melalui proses normalisasi seperti menghapus karakter khusus, mengubah kata menjadi lowercase, menghapus symbol, lematisasi, dan menghapus stopword. Data set langsung di train ke model.

```
In [59]: base_model = LogisticRegression(random_state = 8)
         base_model.fit(features_train, labels_train)
         accuracy_score(labels_test, base_model.predict(features_test))

         C:\Users\mmukhtar\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\l
         ult solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
           FutureWarning)
         C:\Users\mmukhtar\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\l
         ult multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence
           "this warning.", FutureWarning)

Out[59]: 0.9161676646706587

In [60]: best_classifier.fit(features_train, labels_train)
         accuracy_score(labels_test, best_classifier.predict(features_test))

Out[60]: 0.9191616766467066
```

(b) Proses tanpa lemmatization

- Data set dilakukan pembersihan terhadap symbol, karakter khusus, huruf 's dan mengubah huruf menjadi lowercase. Setelah itu preprocessing langsung ke tahap penghapusan stopwords tanpa proses lemmatization.

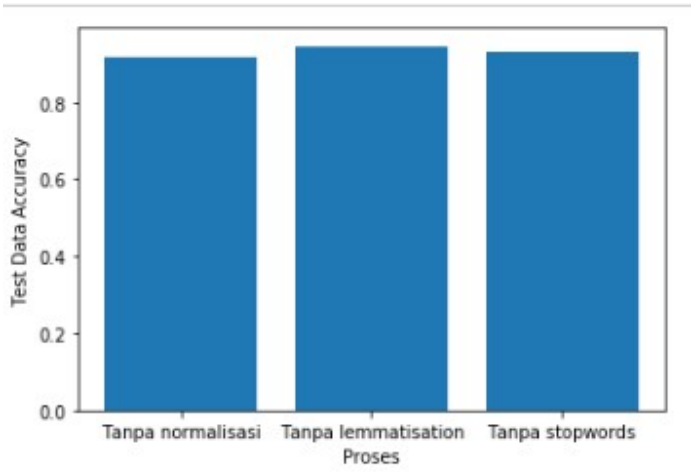
Bandingkan performansi dengan base model, yaitu model dengan parameter default.

```
In [146]: base_model = LogisticRegression(random_state = 8)
          base_model.fit(features_train, labels_train)
          accuracy_score(labels_test, base_model.predict(features_test))

          C:\Users\mmukhtar\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning: Defa
          ult solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
            FutureWarning)
          C:\Users\mmukhtar\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:469: FutureWarning: Defa
          ult multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
            "this warning.", FutureWarning)

Out[146]: 0.9461077844311377

In [147]: best_classifier.fit(features_train, labels_train)
          accuracy_score(labels_test, best_classifier.predict(features_test))

Out[147]: 0.9461077844311377
```

(c) Data set dilakukan pembersihan terhadap symbol, karakter khusus, huruf 's dan mengubah huruf

menjadi lowercase. Setelah itu preprocessing ke tahap lemmatization dan tanpa melakukan proses penghapusan stopword.

```
In [120]: base_model = LogisticRegression(random_state = 8)
          base_model.fit(features_train, labels_train)
          accuracy_score(labels_test, base_model.predict(features_test))

          C:\Users\mmukhtar\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logist
          ult solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
            FutureWarning)
          C:\Users\mmukhtar\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logist
          ult multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this
            "this warning.", FutureWarning)

Out[120]: 0.9221556886227545

In [121]: best_classifier.fit(features_train, labels_train)
          accuracy_score(labels_test, best_classifier.predict(features_test))

Out[121]: 0.9311377245508982
```




Terlihat dari akurasi yang dihasilkan dalam klasifikasi teks tanpa melakukan salah satu atau semua pre-processing bisa disimpulkan:

1. Jika tidak dilakukan pre-processing maka model klasifikasi menjadi lebih buruk karena data set belum dinormalisasi agar seragam (huruf kecil, tanpa symbol, lemmatization, hapus stopword, dll) saat diklasifikasi.

2. Hasil model yang tanpa melakukan proses lemmatization lebih baik daripada model tanpa proses penghapusan stopword. Ini terjadi karena tanpa proses penghapusan stopword, maka kata yang dihasilkan dari dataset masih terdapat stopword yang tidak terlalu memiliki arti lebih sehingga proses klasifikasi menjadi lebih buruk.

3. Tanpa proses lemmatization, kata yang dihasilkan tidak terdapat stopword, sehingga proses klasifikasi menjadi lebih baik walaupun kata dalam kalimat yang belum diubah menjadi kata dasar, karena kata-kata yang keluar hampir sama jadi tetap saja akan terklasifikasi.

4. Yang paling berpengaruh adalah proses penghapusan stopword.

**3. Coba buat tfidf dengan nilai "max_features" yang berbeda-beda (lebih besar dan lebih kecil dari 300), lalu laporkan pengaruhnya terhadap akurasi klasifikasi!**

Jawab :

Yang dilakukan adalah pre-processing terlebih dahulu sampai tahap penghapusan stopword. Lalu saat membuat tf-idf, max_features diubah menjadi 200, 300 dan 400

(a) max_features = 200

```
In [41]: base_model = LogisticRegression(random_state = 8)
         base_model.fit(features_train, labels_train)
         accuracy_score(labels_test, base_model.predict(features_test))

         C:\Users\mmukhtar\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\li
         ult solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this
           FutureWarning)
         C:\Users\mmukhtar\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\li
         ult multi_class will be changed to 'auto' in 0.22. Specify the multi_class optio
           "this warning.", FutureWarning)

Out[41]: 0.9251497005988024

In [42]: best_classifier.fit(features_train, labels_train)
         accuracy_score(labels_test, best_classifier.predict(features_test))

Out[42]: 0.9191616766467066
```

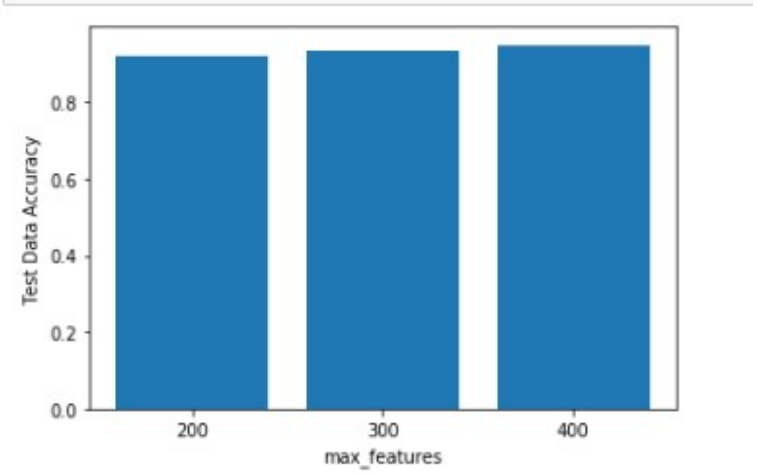
(b) max_features = 300

```
n [61]: base_model = LogisticRegression(random_state = 8)
        base_model.fit(features_train, labels_train)
        accuracy_score(labels_test, base_model.predict(featu

        C:\Users\mmukhtar\AppData\Local\Continuum\anaconda3\
        ult solver will be changed to 'lbfgs' in 0.22. Speci
          FutureWarning)
        C:\Users\mmukhtar\AppData\Local\Continuum\anaconda3\
        ult multi_class will be changed to 'auto' in 0.22. S
          "this warning.", FutureWarning)

ut[61]: 0.9401197604790419

n [62]: best_classifier.fit(features_train, labels_train)
        accuracy_score(labels_test, best_classifier.predict(

ut[62]: 0.9341317365269461
```

(c) max_features = 400

```
In [18]: base_model = LogisticRegression(random_state = 8)
         base_model.fit(features_train, labels_train)
         accuracy_score(labels_test, base_model.predict(features_test))

         C:\Users\mmukhtar\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432:
         ult solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
           FutureWarning)
         C:\Users\mmukhtar\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:469:
         ult multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
           "this warning.", FutureWarning)

Out[18]: 0.9461077844311377

In [19]: best_classifier.fit(features_train, labels_train)
         accuracy_score(labels_test, best_classifier.predict(features_test))

Out[19]: 0.9491017964071856
```

Kesimpulan :




Dari grafik diatas, terlihat semakin banyak max_features dari tf-idf maka akurasi yang dihasilkan semakin baik. Jika dianalisis, ini terjadi karena semakin banyak fitur sebagai informasi pada model maka akan semakin baik pula model klasifikasinya.

**4. Coba dengan beberapa algoritma klasifikasi yang berbeda (minimal 2 algoritma), carilah parameter terbaik (jelaskan nilai2 parameter yang telah dicoba untuk tiap jenis algoritma).**

Jawab :

Semua percobaan akan dilakukan pre-processing terlebih dahulu lalu menggunakan parameter tf-idf yang sama agar hasil yang dianalisis hanya model algoritmanya saja.

Pencarian hyperparameter pertama memakai Randomize Grid Search dengan cross validation agar tidak terlalu makan waktu yang banyak. Setelah dapat hasilnya, parameter yang ditemukan Randomize Search akan digunakan pada Grid Search cross validation dengan penyesuaian pencarian hyperparameter lain yang ingin dicari untuk mendapatkan hasil yang lebih maksimal.

Nilai hyperparameter yang dicari pada model adalah masing2 kombinasi dari parameter yang dimasukan (berdasarkan input array masing2).

(a) Model Logistic Regression

```
random_grid = {'C': [float(x) for x in np.linspace(start = 0.1, stop = 1.9, num = 10)],
               'multi_class': ['multinomial'],
               'solver': ['newton-cg', 'sag', 'saga', 'lbfgs'],
               'class_weight': ['balanced', None],
               'penalty': ['l2']}
```

```
The best hyperparameters from Random Search are:
{'solver': 'newton-cg', 'penalty': 'l2', 'multi_class': 'multinomial', 'class_weight': 'balanced', 'C': 1.7}

The mean accuracy of a model with these hyperparameters is:
0.960867265996827
```

```
The best hyperparameters from Grid Search are:
{'C': 1.7, 'class_weight': 'balanced', 'multi_class': 'multinomial', 'penalty': 'l2', 'solver': 'newton-cg'}

The mean accuracy of a model with these hyperparameters is:
0.9685333333333334
```

```
In [57]: # base_model = LogisticRegression(random_state = 8)
         base_model = classifier
         base_model.fit(features_train, labels_train)
         accuracy_score(labels_test, base_model.predict(features_test))

         C:\Users\mmukhtar\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_mod
         ult solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
           FutureWarning)
         C:\Users\mmukhtar\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_mod
         ult multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to sil
           "this warning.", FutureWarning)

Out[57]: 0.9401197604790419

In [59]: best_classifier.fit(features_train, labels_train)
         accuracy_score(labels_test, best_classifier.predict(features_test))

Out[59]: 0.937125748502994
```

Setelah dicari didapatkan parameter untuk linear regression yaitu dengan nilai ‘C’ sebagai nilai regulasi (semakin kecil semakin bagus). ‘class_weight = balance’ untuk menyesuaikan bobot frekuensi dalam data input. ‘multi_class = multinomial’ sebagai nilai kerugian yang diminimalkan berdasarkan seluruh distribusi probabilitas. ‘penalty = l2’ aturan dalam menghitung penalty adalah l2. Dan ‘solver = newton-cg’ berarti algoritma newton-cg akan digunakan untuk mengoptimasi model logistic regression.

(b) Model Support Vector Machine

```
random_grid = {'C': [.0001, .001, .01],
               'kernel': ['linear', 'rbf', 'poly'],
               'gamma': [.0001, .001, .01, .1, 1, 10, 100],
               'degree': [1, 2, 3, 4, 5],
               'probability': [True]
              }
```

```
The best hyperparameters from Random Search are:
{'probability': True, 'kernel': 'poly', 'gamma': 10, 'degree': 4, 'C': 0.01}

The mean accuracy of a model with these hyperparameters is:
0.9254362771020624
```

```
The best hyperparameters from Grid Search are:
{'C': 0.01, 'degree': 4, 'gamma': 10, 'kernel': 'poly', 'probability': True}

The mean accuracy of a model with these hyperparameters is:
0.9152
```

```
base_model = classifier
base_model.fit(features_train, labels_train)
accuracy_score(labels_test, base_model.predict(features_test))
```

```
C:\Users\mmukhtar\AppData\Local\Continuum\anaconda3\lib\site-packages\sklea
of gamma will change from 'auto' to 'scale' in version 0.22 to account bett
uto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
```

Out[33]: 0.2155688622754491

In [34]:
```
best_classifier.fit(features_train, labels_train)
accuracy_score(labels_test, best_classifier.predict(features_test))
```

Out[34]: 0.9251497005988024

Penjelasan parameter : ‘C’ sebagai parameter regulasi, harus angka positif. ‘kernel’ sebagai aturan algoritma untuk mengola data. ‘gamma’ sebagai koefisien untuk kernel. ‘degree’ untuk parameter koefisien kernel ‘poly’. ‘probabilty’ jika dihidupkan maka pencarian akan menggunakan 5-fold-cross-validation.

(c) Random Forest Classifier

```
The best hyperparameters from Random Search are:
{'n_estimators': 1000, 'min_samples_split': 10, 'min_samples_leaf': 1, 'max_features': 'auto', 'max_depth': 100, 'bootstrap': F
alse}

The mean accuracy of a model with these hyperparameters is:
0.9428873611845584

The best hyperparameters from Grid Search are:
{'bootstrap': False, 'max_depth': 100, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators':
1000}

The mean accuracy of a model with these hyperparameters is:
0.9424
```

In [22]:
```
# base_model = LogisticRegression(random_state = 8)
base_model = classifier
base_model.fit(features_train, labels_train)
accuracy_score(labels_test, base_model.predict(features_test))
```

```
C:\Users\mmukhtar\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: FutureWarning: The defaul
t value of n_estimators will change from 10 in version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```

Out[22]: 0.8952095808383234

In [21]:
```
best_classifier.fit(features_train, labels_train)
accuracy_score(labels_test, best_classifier.predict(features_test))
```

Out[21]: 0.9281437125748503

Penjelasan hyperparameter : n_estimator adalah jumlah tree dari model yang ingin dibuat (berarti ada 1000 pohon dalam model ini). min_samples_split adalah minimum sample split dari nodenya (berarti setiap node minimal ada 10 data). ‘min_samples_leaf’ adalah minimum sample data pada setiap leaf node (berarti setiap leaf minimum daunnya 1). ‘max_features = auto’ adalah aturan untuk mencari best split. ‘max_depth’ adalah maximum kedalaman tree (berarti maximum kedalaman tree adalah 100). ‘bootstrap = False’ berarti semua dataset akan dimasukan ke setiap tree.

**Kesimpulan** :

1. Dari akurasi yang didapatkan, model Logistic Regression lebih baik daripada model yang lain karena pada model ini terdapat parameter multi_class yang menyebabkan model bisa memprediksi class

banyak sesuai dengan class yang ada. Lalu ada juga parameter class_weight “balance” yang menyebabkan bobot antar class menjadi seimbang. Dan pada model ini data dipisahkan menggunakan garis.

2. Terlihat juga perbandingan terhadap base model classifier yang belum ditemukan hyperparameternya akan menjadi lebih buruk dibandingkan dengan model yang memakai hyperparameter.

3. Base model classifier logistic regression lebih baik jika dibandingkan dengan model logistic regression memakai hyperparameter. Ini terjadi karena pada proses pencarian akurasi hyperparameter menggunakan randomize dan grid search memakai rata-rata akurasi hasil cross validation. Jadi bisa saja model yang dihasilkan lebih baik atau lebih buruk.

**5. Jika anda ingin menggunakan teks bahasa Indonesia, bagian mana saja yang perlu dilakukan penyesuaian?**

Jawab :

Ada dua cara:

1. Mengubah Bahasa Indonesia menjadi Bahasa inggris dengan men-translatenya. Sehingga proses yang dilakukan sama seperti pada Bahasa inggris. Tetapi teorinya akan lebih jelek model klasifikasinya karena data bahasa inggris yang ditranslate bisa saja salah terjemahan.

2. Memakai Bahasa Indonesia : melakukan pre-processing pada tahap menghilangan symbol, mengecilkan huruf (case folding) seperti biasa. Lalu pada lemmatization dibutuhkan lemmatizer bahasa Indonesia dan penghilangan stopword juga membutuhkan data stopword dari kata berbahasa Indonesia.