LAPORAN TASK KLASIFIKASI

IF5153 Natural language processing

DISUSUN OLEH

M FARREL DANENDRA RACHIM - 13521048

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

2024

DAFTAR ISI

BAB I

PENJELASAN KODE

Tugas ini mengimplementasikan *feature extraction* berupa bag of words dengan dengan tiga traditional ML algorithm yang berbeda. Kode ini melakukan preprocessing berupa menghilangkan tanda baca pada dataset yang memiliki teks dan label sentimen untuk setiap baris teks tersebut. Teks sudah di-*lowercase* dalam datasetnya.

Berikut adalah link github kode: https://github.com/Breezy-DR/Classification-Task-13521048

Terdapat tiga file kode yang berbeda, masing-masing memiliki model ML tradisional yang berbeda, yaitu Decision Tree, Naive Bayes, dan Support Vector Machine (SVM). Walaupun begitu, masing-masing file memiliki struktur kode yang sama, yakni import library, pembacaan data, preprocessing, exploratory data analysis (EDA), encoding label, serta skenario-skenario eksperimen yang dijalankan.

1.1. Import Library

Kode ini menggunakan library pandas untuk membaca data, numpy dan matplotlib untuk melakukan EDA, re untuk preprocessing, serta scikit-learn untuk membangun model.

1.2. Pembacaan Data

Terdapat empat file data yang digunakan untuk membangun model ini, yaitu training data, validation data, testing data dengan label yang di-mask, dan testing data dengan label yang di-unmask. Untuk keperluan perbandingan hasil label dengan pelatihan model, saya hanya akan memanfaatkan unmasked testing data. Selain itu, ada juga dokumen vocabulary yang berisi seluruh token data. Semua file terletak di dalam folder sentiment-prosa di repository Github. Untuk menjalankan kode, download folder tersebut dan unggah ke Google Drive untuk memperoleh izin *mounting*. Dalam proses ini juga dilakukan penamaan kolom data yang akan digunakan.

1.3. Preprocessing

Data di-preprocess menggunakan regular expression (re) untuk menghilangkan tanda baca yang masih tersisa dalam data serta menghilangkan whitespace dari setiap token. Selain itu, juga dilakukan penghilangan *stop words* bahasa Indonesia dan tokenisasi dengan library NLTK.

1.4. Exploratory Data Analysis (EDA)

Saya membuat grafik bar chart untuk melihat distribusi data berdasarkan label sentimennya: negative, neutral, atau positive. Ternyata, data yang memiliki sentimen positif jauh lebih banyak daripada sentimen negatif atau netral.

1.5. Encoding Label

Label akan di-encode menjadi 0, 1, dan 2 agar pelatihan model berjalan dengan lancar.

1.6. Skenario Eksperimen

Dalam setiap skenario eksperimen, akan dilakukan proses training dan evaluasi model training menggunakan precision, recall, accuracy, dan F1-score, baik dalam data validasi maupun dalam data testing.

1.6.1. Training

Pertama-tama Bag Of Words (BoW) dibuat menggunakan CountVectorizer, di mana token-token digolongkan berdasarkan frekuensi penggunaannya. Vectorizer kemudian dihilangkan jika ada isi duplikat dan dilakukan transformasi terhadap data-data yang tersedia.

Lalu, model algoritma dibuat dan data latih yang sudah di-transform akan di-fit.

1.6.2. Evaluasi Model Training

Setelah itu, akan diukur performa model menggunakan data validasi dan data tes. Kedua data akan diprediksi dan diukur menggunakan skor akurasi, precision, recall, dan F1 score, serta classification report. Selain itu, akan dihitung juga banyak data testing yang salah diprediksi (hasil prediksi model tidak sama dengan label asli data testing), serta dibuat grafik yang menunjukkan distribusi deteksi error dari hasil prediksi.

BAB II

SKENARIO EKSPERIMEN

2.1. Decision Tree

Model pertama yang akan diuji adalah *decision tree* yang diterapkan melalui model DecisionTreeClassifier dari library scikit learn dengan parameter random_state=42. Selain itu, dilakukan juga *hyperparameter tuning* menggunakan *grid search* oleh GridSearchCV dan scoring “f1_macro” untuk parameter 'max_depth': [5, 10, 20, 50, 100, 300], 'min_samples_split': [2, 5, 10], dan 'min_samples_leaf': [2, 5, 10]. Parameter terbaik yang terpilih adalah 'max_depth': 300, 'min_samples_leaf': 5, dan 'min_samples_split': 2.

2.2. Naive Bayes

Model kedua yang akan diuji adalah Naive Bayes. Terdapat dua jenis model Naive Bayes dalam kasus ini, Multinomial Naive Bayes dan Gaussian Naive Bayes. Selain itu, dilakukan juga *hyperparameter tuning* khusus untuk Multinomial Naive Bayes menggunakan *grid search* oleh GridSearchCV dan scoring “f1_macro” untuk parameter 'alpha': [0.1, 0.5, 0.75, 1.0, 2.0]. Parameter terbaik yang terpilih adalah 'alpha': 0.5.

2.3. Support Vector Machine (SVM)

Model ketiga yang akan diuji adalah SVM. Terdapat tiga jenis kernel model SVM dalam kasus ini, linear, *radial basis kernel* (RBF), dan poly. Selain itu, dilakukan juga *hyperparameter tuning* menggunakan *grid search* oleh GridSearchCV dan scoring “f1_macro” untuk parameter 'kernel': ['linear', 'rbf'], 'C': [0.1, 1, 10, 20]. Parameter terbaik yang terpilih adalah 'C': 10, 'kernel': 'rbf'.

BAB III

HASIL EKSPERIMEN

3.1. Hasil Data Validasi

Berikut adalah tabel skor akurasi, precision, recall, dan F1 score untuk data validasi dari masing-masing eksperimen.

	Akurasi	Precision	Recall	F1 score
Decision Tree	0.776984	0.717879	0.715166	0.715960
Hyperparameter DT	0.767460	0.713337	0.697577	0.700946
Multinomial NB	0.850794	0.843290	0.795259	0.814817
Gaussian NB	0.652381	0.602822	0.626843	0.604068
Hyperparameter NB	0.857937	0.835911	0.826003	0.830462
SVM linear kernel	0.829365	0.803813	0.765893	0.780240
SVM RBF kernel	0.844444	0.844701	0.736094	0.763456
SVM poly kernel	0.671429	0.809460	0.531121	0.479614
Hyperparameter SVM	0.846031	0.829570	0.777244	0.798271

Tabel 3.1.1. Data Validasi

Model hyperparameter NB memiliki nilai akurasi, recall, dan F1 score paling tinggi dibandingkan model-model lain, sedangkan model SVM RBF kernel memiliki nilai precision paling tinggi.

3.2. Hasil Data Testing

Berikut adalah tabel skor akurasi, precision, recall, dan F1 score untuk data testing dari masing-masing eksperimen.

	Akurasi	Precision	Recall	F1 score
Decision Tree	0.53	0.528150	0.487548	0.497557
Hyperparameter DT	0.582	0.562374	0.515112	0.515158
Multinomial NB	0.654	0.658025	0.597722	0.598694
Gaussian NB	0.504	0.468687	0.469843	0.454559
Hyperparameter NB	0.638	0.646702	0.608877	0.599047
SVM linear kernel	0.71	0.697164	0.665815	0.675722
SVM RBF kernel	0.664	0.723846	0.566274	0.559109
SVM poly kernel	0.456	0.425583	0.371669	0.276163
Hyperparameter SVM	0.716	0.693028	0.659759	0.668897

Tabel 3.2.1. Data Testing

Model hyperparameter SVM memiliki nilai akurasi testing paling tinggi, SVM RBF kernel memiliki nilai precision testing paling tinggi, dan SVM linear kernel memiliki nilai recall dan F1 score testing terbesar.

BAB IV

ERROR ANALYSIS

Analisis error akan dihitung melalui banyak data yang salah diprediksi dari data testing, dan distribusi error untuk setiap label dengan bar chart.

	Banyak Error	Distribusi Data
Decision Tree	235	text emotion negative 81 neutral 61 positive 93
Hyperparameter DT	209	text emotion negative 43 neutral 68 positive 98
Multinomial NB	173	text emotion negative 15 neutral 57 positive 101
Gaussian NB	248	text emotion negative 51 neutral 60 positive 137
Hyperparameter NB	181	text emotion negative 17 neutral 46 positive 118

SVM linear kernel	145	text emotion negative 41 neutral 46 positive 58
SVM RBF kernel	168	text emotion negative 19 neutral 75 positive 74
SVM poly kernel	272	text emotion negative 4 neutral 88 positive 180
Hyperparameter SVM	142	text emotion negative 39 neutral 51 positive 52

Tabel 4.1. Error Analysis

Model yang memiliki data error paling sedikit adalah hyperparameter SVM sebanyak 142, dengan data dengan error paling banyak berupa label neutral dan positive. Model yang memiliki data error paling banyak adalah SVM poly kernel sebanyak 272, dengan data dengan error paling banyak berupa label positive sebanyak 180.