Classifying Wine Varieties Based on The Provided Wine Expert Descriptions in The Wine Reviews Dataset using DistilBERT as the Language Model

Dosen Pengampu Mata Kuliah
Lili Ayu Wulandhari, S.Si.,M.Sc.,Ph.d.

Disusun Oleh:
2501972493 - Gaizkia Adeline Atmaka
2502003895 - Glory Daniella
2540133000 - Jeremy Saputra Tatuil
2501972625 - Shelly Alfianda

Kelas:
LA05

Jurusan Teknik Informatika dan Matematika
Fakultas Teknik Informatika
Universitas Bina Nusantara
2024

Daftar Isi

I. Pendahuluan

Wine merupakan minuman yang memiliki beragam variasi, baik dari jenis anggur yang digunakan, proses pembuatan, hingga tempat asalnya. Keragaman ini menciptakan beragam jenis wine dengan karakteristik berbeda yang sulit dibedakan satu sama lain oleh konsumen. Sistem klasifikasi yang akurat diperlukan untuk membantu konsumen memahami wine dan memilih sesuai dengan selera mereka.

Pendekatan yang biasa dipakai untuk mengklasifikasikan jenis wine yaitu dengan memanfaatkan penjelasan tekstual yang biasanya disertai produk wine. Penjelasan ini mencakup berbagai aspek seperti aroma, rasa, kekuatan, dan asal daerah yang bisa memberikan petunjuk tentang jenis wine tersebut. Tetapi, penjelasan ini seringkali memakai bahasa yang sulit untuk dipahami, sehingga analisis manual menjadi kurang efektif dan efisien.

Ekstraksi informasi dari teks yaitu deskripsi yang menyertai setiap data wine dilakukan, sehingga informasi tersebut dapat digunakan untuk mengklasifikasikan jenis wine berdasarkan deskripsinya. Untuk melakukan hal ini secara otomatis, kita memerlukan model NLP (Natural Language Processing) yang dapat memahami dan mengekstraksi informasi penting dari teks deskripsi tersebut.

DistilBERT adalah model yang cocok untuk tugas ini karena kemampuannya untuk memahami konteks dalam teks memiliki efisiensi yang lebih tinggi dibandingkan model BERT (Bidirectional Encoder Representations from Transformers). DistilBERT adalah versi terdistilasi dari BERT yang lebih ringan dan cepat namun tetap mempertahankan kemampuan pemahaman teksnya.

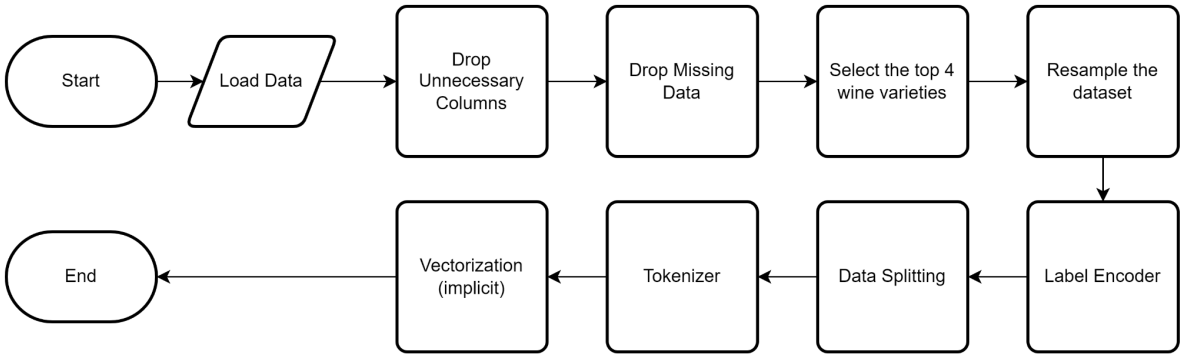
Data yang digunakan berasal dari Kaggle dengan judul *Wine Reviews*. Hasil analisa sederhana terhadap data menunjukkan varietas yang sangat besar pada kolom *wine variety*, sehingga diambil empat jenis *wine* dengan jumlah data terbanyak. Data testing akan diklasifikasikan ke dalam empat jenis *wine* tersebut berdasarkan kolom *description*.

II. Metodologi

Exploratory Data Analysis

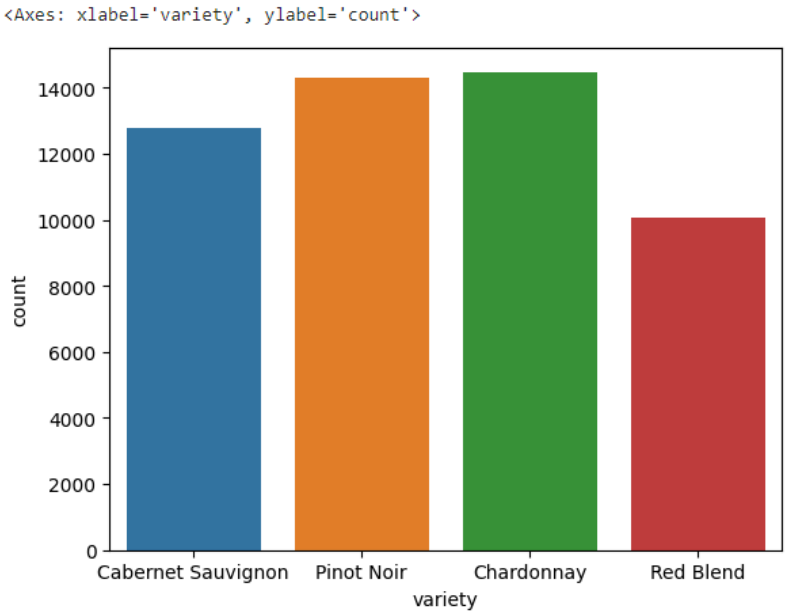
Sebelum memproses data, dilakukan *Exploratory Data Analysis* terlebih dahulu. Pada bagian ini dataset akan dilakukan pengecekan seperti : memeriksa info dataset, *missing* data, *unique* value, dan visualisasi. Hal ini perlu dilakukan untuk membantu mendapatkan wawasan serta informasi dari data tersebut. Sehingga, dapat terlihat distribusi data yang dimiliki, yang nantinya akan mempermudah step berikutnya yaitu preprocessing.

Preprocessing

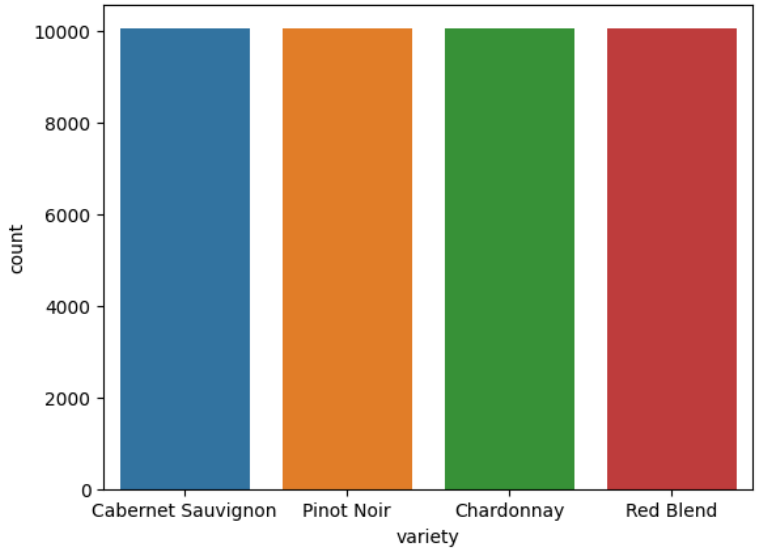


Dari *Wine Reviews Dataset*, data akan diambil dari *file csv "winemag-data_first_150k.csv*", Dalam dataset fitur yang dibutuhkan hanyalah data yang terdapat pada kolom *description* (teks yang berisikan penjelasan rasa dari anggur yang diulas) dan *variety* (jenis anggur yang diulas), sehingga fitur/kolom lain akan dibuang. Karena jumlah data yang terdapat dalam dataset banyak, maka data yang tidak ada nilai pada kolom *variety* akan dibuang.

Dalam dataset tersebut terdapat 632 *class* pada fitur/kolom *variety* (jenis anggur yang diulas) dengan jumlah data yang tidak seimbang (ada *class* yang memiliki banyak data dan ada *class* yang memiliki sedikit data), maka dipilih empat *class* pertama yang memiliki data yang paling banyak, dan dari keempat *class* tersebut diambil jumlah data terkecil dan tiga *class* sisanya akan disampling sejumlah nilai data tersebut, sehingga keempat *class* tersebut memiliki jumlah data yang sama.



Empat *class variety* dengan jumlah data terbanyak



Empat *class variety* dengan jumlah data terbanyak (setelah disampling)

Vectorization

Karena model *classifier* yang akan dibuat berbasiskan model bahasa DistilBERT, maka pada tahap *vectorization* metode *vectorization* yang digunakan adalah dengan menggunakan *tokenizer* yang sama pada saat model DistilBERT dilakukan *pre-training*.

Tokenizer (serta model) dapat diperoleh menggunakan library Hugging Face, input dari *tokenizer* tersebut adalah sebuah kalimat dan output dari tokenizer adalah *input ids* dan *attention mask*, *input_ids* adalah sebuah array berisikan kumpulan *identifier* unik yang ditetapkan pada masing-masing *token* yang akan digunakan oleh

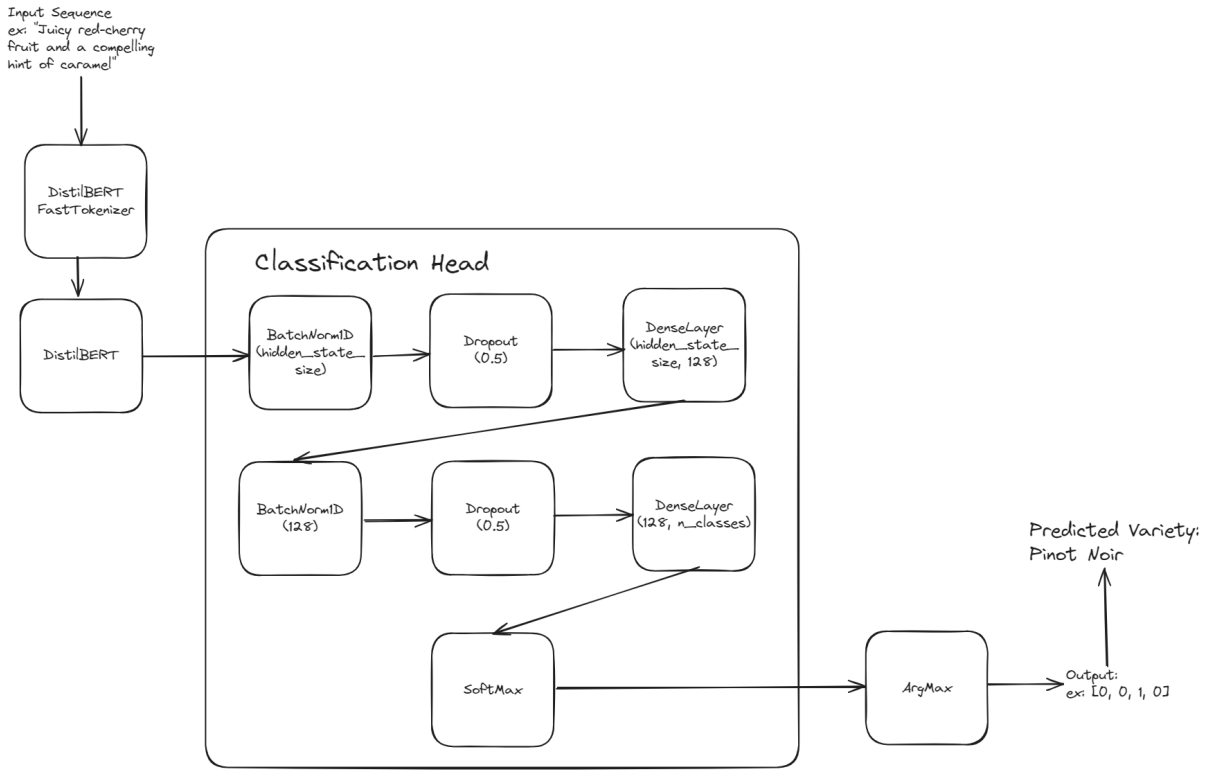
embedding layer pada model untuk diubah menjadi sebuah vektor, sedangkan *attention mask* adalah kumpulan angka nol atau satu, yang menandakan apakah token, pada indeks tersebut perlu dilakukan komputasi pada *attention mechanism* yang ada dalam model, apabila perlu dilakukan maka pada *attention mask* pada indeks tersebut akan bernilai satu, jika tidak nol.

Modeling and Parameter Tuning

Untuk menghasilkan sebuah model *classifier* yang mampu memahami makna teks, maka digunakanlah model DistilBERT (**uncased**, tidak membedakan huruf *lowercase* dengan *uppercase*), yang merupakan turunan dari model utama yang lebih besar yaitu BERT, model ini terpilih karena ukurannya yang kecil (jumlah parameter yang lebih sedikit dibandingkan dengan BERT), kecepatan *inference* yang lebih cepat, efisiensinya, dan mampu memiliki 95% performa dari BERT.

Dikarenakan data-data yang digunakan pada saat *pre-training* model DistilBERT adalah data-data teks yang tidak diperlakukan *word filtering*, *stemming*, dan *lemmatization*, maka pada tahap preprocessing tidak diperlakukan langkah-langkah tersebut juga, sehingga data yang diproses (pada saat *fine-tuning*) memiliki bentuk kurang lebih sama seperti data yang diproses pada saat *pre-training*. Hal ini tidak dilakukan karena model BERT mengandalkan *subword tokenization* sehingga kata-kata akan dipisah lebih lanjut dan digunakan pada tahap *attention*.

Output dari model merupakan kumpulan *dense vector* untuk masing-masing token yang telah diproses oleh DistilBERT yang disebut sebagai *hidden state*, dalam *hidden state* terdapat vektor untuk token [CLS] (*Classification token* adalah token pertama dalam *hidden state*) yang merupakan sebuah *pooled output* yang merepresentasikan seluruh *token* dalam *sequence* dalam satu buah *context vector*, vektor ini dapat kita gunakan sebagai input dalam *layer-layer* klasifikasi (*classification head*). Maka arsitektur akhir dari model yang digunakan adalah sebagai berikut:



Parameter-tuning dilakukan secara manual (karena keterbatasan waktu), dan hyperparameter yang diubah dan diuji adalah *learning rate* (1e-3 dengan 1e-5), jumlah *epoch* (5 dan 10), *batch size* (16 dan 40), *nilai dropout* (0.3 dan 0.5), dan dimensi *hidden dense layer* (256 dan 128). Dan untuk *loss function* yang digunakan adalah *cross entropy loss function* dan *optimizer function* yang digunakan adalah *Adam*.

III. Hasil dan Analisa

Setelah dilakukan *fine-tuning* dan *parameter-tuning* diperoleh *hyperparameter* yang paling optimal adalah sebagai berikut:

- *learning rate* = 1e-5
- *epoch* = 10
- *batch size* = 16
- *dropout rate* = 0.5
- *hidden dense layer dimension* = 128

Hasil *classification report* model adalah sebagai berikut:

	precision	recall	f1-score	support
Cabernet Sauvignon	0.82	0.90	0.86	1027
Chardonnay	0.99	0.98	0.98	997
Pinot Noir	0.94	0.87	0.91	1002
Red Blend	0.89	0.88	0.88	999
accuracy			0.91	4025
macro avg	0.91	0.91	0.91	4025
weighted avg	0.91	0.91	0.91	4025

Didapatkan hasil accuracy sebesar 91%, dimana hasil tersebut menunjukkan bahwa model dapat mengklasifikasikan wine dengan performa yang baik. Performa untuk setiap jenis wine cukup konsisten, dengan class Chardonnay yang memiliki performa terbaik dalam hal precision dan recall juga f1-score. Performa model menunjukkan bahwa model andal dalam mengklasifikasikan jenis wine berdasarkan deskripsi yang diberikan.

IV. Kesimpulan

Pengujian klasifikasi jenis wine berdasarkan deskripsinya pada penelitian ini menggunakan model DistilBERT. Model ini digunakan dikarenakan ukurannya lebih kecil, kecepatan *inference* yang lebih cepat, efisiensinya, dan mampu memiliki 95% performa dari BERT. Pengujian ini juga belum maksimal dikarenakan parameternya masih ditentukan manual seharusnya bisa dicari parameter yang paling baik untuk digunakan. Untuk hasil akhirnya, model DistilBERT menghasilkan accuracy sebesar 91%, yang artinya model dapat mengklasifikasikan jenis wine berdasarkan description dengan baik.

V. Lampiran

Berikut adalah link dasaset yang digunakan:
https://www.kaggle.com/datasets/zynicide/wine-reviews

Berikut adalah link video presentasi yang berisi penjelasan:
https://www.youtube.com/watch?v=hq0O6I6SZnA

Berikut adalah link code untuk tugas ini :
https://drive.google.com/file/d/1U8jZH6g9jlLIO29woHomU_Azv3oE7Ue3/view?usp=drive_link

VI. Referensi

https://huggingface.co/distilbert/distilbert-base-uncased

Victor SANH, Lysandre DEBUT, Julien CHAUMOND, Thomas WOLF. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. https://arxiv.org/pdf/1910.01108v4

Victor Sanh. 2019. Smaller, faster, cheaper, lighter: Introducing DistilBERT, a distilled version of BERT. https://medium.com/huggingface/distilbert-8cf3380435b5

Vyacheslav Efimov. 2023. Large Language Models: DistilBERT — Smaller, Faster, Cheaper and Lighter. https://towardsdatascience.com/distilbert-11c8810d29fc