Project UTS - Pengolahan Bahasa Alami

KLASIFIKASI TEKS BAHASA INDONESIA UNTUK ...

Anggota Kelompok:

1. Klasik S.K.M.Taidi - 210711185
2. Bima Wahyu L - 210711339
3. Valent Wilbert - 210711340

Objective

Meningkatkan minat dan retensi membaca masyarakat Indonesia adalah tujuan yang penting. Diperlukan upaya kolaboratif dari pemerintah, lembaga pendidikan, dan masyarakat secara luas. Pengembangan program literasi yang menarik dan mudah diakses dapat menjadi langkah yang efektif dalam mencapai tujuan tersebut.

Latar Belakang

Klasifikasi genre buku memiliki peranan penting dalam industri literatur, tidak hanya membantu pembaca menemukan buku sesuai minat mereka tetapi juga mendukung penerbit dalam memahami pasar target dan merancang strategi pemasaran. Namun, tantangan muncul karena banyak buku yang mencampur unsur-unsur genre, dan batasan antara genre seringkali menjadi kabur. Oleh karena itu, penting untuk mengembangkan metode klasifikasi yang akurat dan efektif guna memahami kompleksitas sastra modern dan memfasilitasi navigasi yang lebih baik di antara berbagai jenis buku, sekaligus mendukung pemahaman yang lebih baik tentang preferensi pembaca dan tren pasar, serta terus memperbarui sistem klasifikasi ini sesuai dengan perkembangan industri literatur.

Deskripsi Dataset

Projek yang kami buat kali ini memanfaatkan dataset dari Google Book API yang kami akses melalui kode Python yang kami buat sendiri, khususnya untuk mengakses buku-buku berbahasa Indonesia. Dengan memanfaatkan API ini, kami dapat mengumpulkan informasi yang relevan untuk analisis dan pengembangan projek kami. Penggunaan sumber data ini menjadi langkah awal yang penting dalam proses merancang dan melaksanakan projek kami, karena kami dapat memastikan keakuratan dan kelengkapan informasi yang kami gunakan serta melakukan pengolahan data secara efisien melalui kode Python. Dataset yang kami gunakan memiliki beberapa unsur, seperti Title (judul), Author(Penulis),

Categories(Genre), dan Description(Sinopsis). pada dataset kami juga memiliki jumlah data sebanyak 344 baris yang kami pilih untuk mendapatkan dataset yang seimbang dan memenuhi aspek yang di butuhkan dalam proses learning projek.

Pengembangan Model Klasifikasi Teks

Inisialisasi

Silakan isi bagian ini dengan mengimport library-library yang akan digunakan.

In [ ]:
```
import numpy as np
import pandas as pd
import modSpellChecker as sc
from contractions import CONTRACTION_MAP
import re
import string
import matplotlib.pyplot as plt
import nltk



from sklearn.metrics import ConfusionMatrixDisplay

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
#from pattern.en import tag
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory

#from normalization import normalize_corpus

from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.pipeline import Pipeline

from sklearn.model_selection import train_test_split

from sklearn.feature_selection import SelectFromModel


from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import SGDClassifier
from sklearn.svm import LinearSVC
from sklearn import metrics
from sklearn.naive_bayes import MultinomialNB
```

Data Preparation

Silakan isi bagian ini dengan script untuk memanggil data yang akan digunakan untuk keperluan pengembangan model klasifikasi teks dan menampilkan contoh data yang telah dipanggil baik yang akan dijadikan sebagai input maupun target-nya (label).

In [ ]:

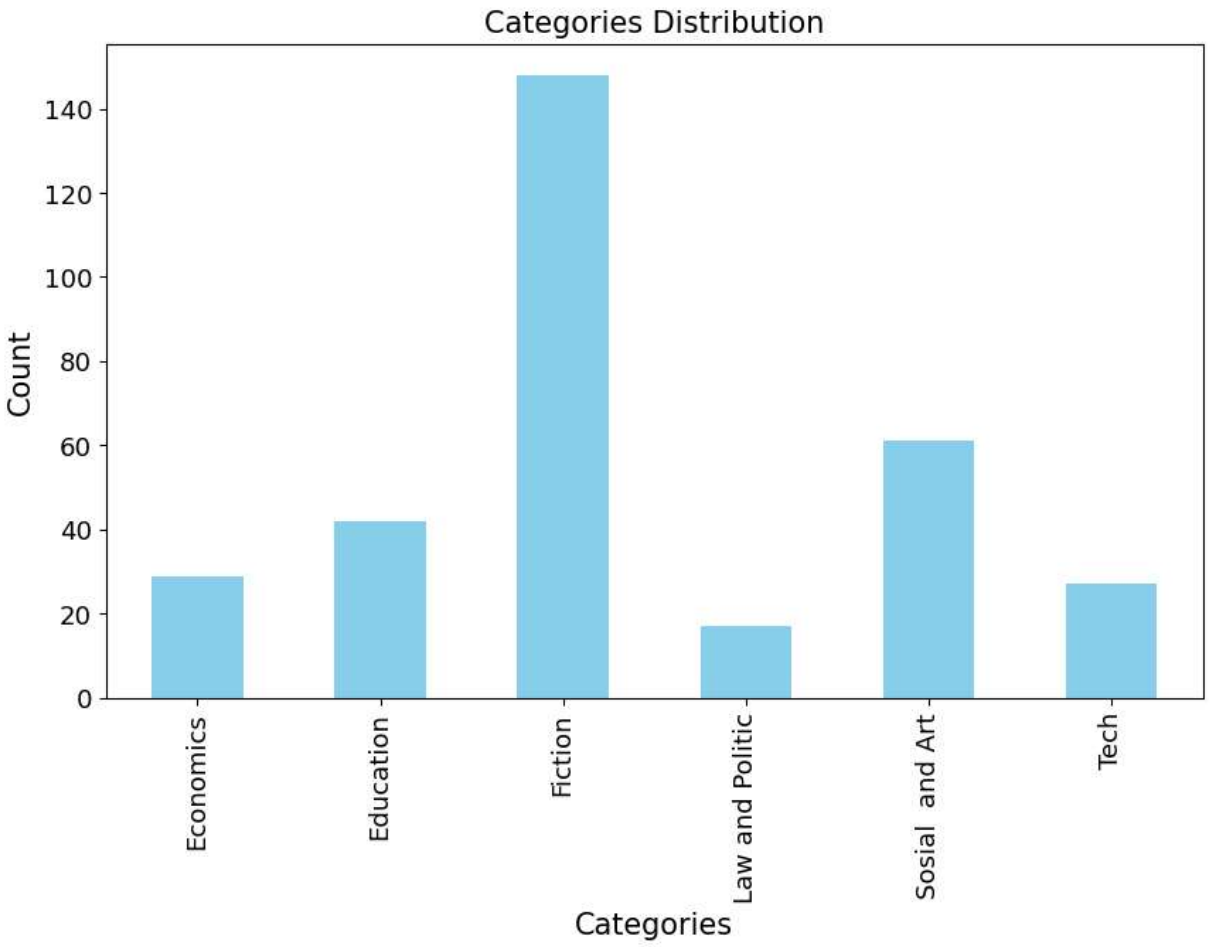
In [ ]:
```
dataset = pd.read_excel('updated_dataset.xlsx')

label = dataset['Categories']
feature = dataset['Description']
```

In [ ]:
```
dataset.shape
```

Out[ ]:
```
(324, 18)
```

In [ ]:
```
df2 = dataset.groupby(['Categories'])['Description'].count()
# create the plot
df2.plot(kind='bar', x='Categories', y='count', figsize=(10,6), color="skyblue", fo

# set the x-label, y-label, and title
plt.xlabel('Categories', fontsize=15)
plt.ylabel('Count', fontsize=15)
plt.title('Categories Distribution', fontsize=15)
```

Out[ ]:
```
Text(0.5, 1.0, 'Categories Distribution')
```



In [ ]:
```
label[0:10]
```

Out[ ]:
```
0    Education
1      Fiction
2    Education
3    Economics
4    Education
5    Economics
6      Fiction
7    Education
8    Education
9    Economics
Name: Categories, dtype: object
```

In [ ]:
```
feature[0:10]
```

Out[ ]:
```
0    Siapa tidak kenal Sisca Soewitomo? Pakar kulin...
1    Ini kisah cinta yang biasa. Tentang tiga orang...
2    Riyas Irmadona, penulis buku ini adalah seoran...
3    Dalam era globalisasi, persaingan usaha sangat...
4    "Pasta adalah makanan olahan yang banyak digu-...
5    "Menjadi Salesman mungkin bukan merupakan pili...
6    Yuna jatuh cinta pada pandangan pertama dengan...
7    Buku ALL IN ONE TOEFL GRAMMAR dirancang secara...
8    Ulangan harian menjadi salah satu instrumen un...
9              Economic development in South Sumatera.
Name: Description, dtype: object
```

Normalisasi Data

Silakan isi bagian ini dengan script untuk keperluan normalisasi data teks yang akan digunakan untuk pengembangan model klasifikasi teks. Untuk setiap strategi normalisasi yang akan digunakan sebaiknya dibuat fungsi tersendiri dan dipanggil dalam satu fungsi normalisasi. Jangan lupa pada bagian ini ditampilkan contoh hasil normalisasi datanya.

In [ ]:
```
character = ['z','y','x','w','v','u','t','s','r','q','p','o','n','m','l','k','j','i
def repeatcharNormalize(text):
  for i in range(len(character)):
    charac_long = 5
    while charac_long>=2:
      char=character[i]*charac_long
      text=text.replace(char,character[i])
      charac_long-=1
      return text

def spellNormalize(text):
    spellCheck = []
    for i in text:
        if i not in character:
            j = sc.correction(i)
            spellCheck.append(j)
        else:
            spellCheck.append(i)
    return spellCheck

def tokenize_text(text):
```

```
  tokens=nltk.word_tokenize(text)
  tokens=[token.strip() for token in tokens]
  return tokens

def expand_contractions(text, contraction_mapping):
    contractions_pattern = re.compile('({})'.format('|'.join(contraction_mapping.ke
                                      flags=re.IGNORECASE|re.DOTALL)
    def expand_match(contraction):
        match = contraction.group(0)
        first_char = match[0]
        expanded_contraction = contraction_mapping.get(match) if contraction_mappin
        expanded_contraction = first_char + expanded_contraction[1:]
        return expanded_contraction

    expanded_text = contractions_pattern.sub(expand_match, text)
    expanded_text = re.sub("'", "", expanded_text)
    return expanded_text

def stemmer_text(text):
  factory = StemmerFactory()
  stemmer = factory.create_stemmer()
  text = stemmer.stem(text)
  return text
```

In [ ]:
```
def remove_special_characters(text):
    tokens = tokenize_text(text)
    pattern = re.compile('[{}]'.format(re.escape(string.punctuation)))
    filtered_tokens = filter(None, [pattern.sub('', token) for token in tokens])
    filtered_text = ' '.join(filtered_tokens)
    return filtered_text

def remove_stopwords(text):
    tokens = tokenize_text(text)
    factory = StopWordRemoverFactory()
    stopword_list = factory.get_stop_words()
    filtered_tokens = [token for token in tokens if token not in stopword_list]
    filtered_text = ' '.join(filtered_tokens)
    return filtered_text

def normalize_corpus(corpus, tokenize=False):
    normalized_corpus = []
    for text in corpus:
        text = expand_contractions(text, CONTRACTION_MAP)
        text = stemmer_text(text)
        text = remove_special_characters(text)
        text = repeatcharNormalize(text)
        text = remove_stopwords(text)
        if tokenize:
            text = tokenize_text(text)
            text = spellNormalize(text)
        normalized_corpus.append(text)
    return normalized_corpus
```

In [ ]:
```
# training dataset
def prepare_datasets(corpus, labels, test_data_proportion=0.3):
```

```
    train_X, test_X, train_Y, test_Y = train_test_split(corpus, labels, test_size=t
    return train_X, test_X, train_Y, test_Y

train_corpus, test_corpus, train_labels, test_labels = prepare_datasets(feature, la
```

In [ ]:
```
# Mengganti nilai-nilai NaN dengan string kosong
train_corpus.fillna('', inplace=True)

# Memanggil fungsi normalize_corpus setelah menangani nilai-nilai yang kosong
norm_train_corpus = normalize_corpus(train_corpus)
```

In [ ]:
```
norm_train_corpus[0:10]
```

Out[ ]: ['puji syukur panjat hadir allah swt atas ken teliti bidang hukum pusat teliti bad
an ahli dewan wakil rakyat republik indonesia dpr ri selesai tulis ilmiah susun be
ntuk buku sambut baik terbit buku lindung penting nasional dagang internasional ba
gai salah satu karya ilmiah hasil teliti bidang hukum pusat teliti badan ahli dpr
ri susun buku dasar hasil teliti tim hukum pusat teliti badan ahli dpr ri laku tah
un 2020 teliti dilatarbelakangi makin tingkat dagang internasional bawa dampak pos
itif dampak negatif penting nasional indonesia dampak negatif sebut singkir usaha
mikro kecil tengah umkm pasar marak edar barang impor bahaya tuduh curang negara m
itra langgar hak kaya intelektual hki lindung penting nasional dampak negatif daga
ng internasional sangat butuh lindung sebut makin rasa penting era pandemi coronav
irues disease 2019 covid19 akibat jadi tingkat dagang cara digital minim lindung e
mpat aspek kaji buku aspek pertama kait hki gambar tulis e epelindungan hak kaya i
ntelektual hki penting nasional da',
 'buku fenomenal a brief history of time stephen hawking tegas ubah cara pikir fis
ika jagat raya realitas lalu buku sebut stephen hawking bagai fisikawan teoretis p
aling cemerlang sejak einstein buka pikir terima gagas ilmiah paling penting dewas
a kosmos sekarang stephen hawking datang bersit cahaya baru kawasan paling gelap r
uangwaktu singkap deret mungkin baru paham jagat raya',
 'e edi arah sana e e si kucing lambai cakar kanan e etinggal si tukang topi belah
sana e e lambai cakar satu e etinggal si kelinci maret serah siapa kunjung gila e
e alice bosan buku sedang baca buku tak gambar maupun cakap ekor kelinci putih lew
at gesa lihat jam saku alice ikut mula tualang alice negeri ajaib e e e enegeri pe
nuh makhluk aneh eksentrik alice temu sang duchess kucing bicara tukang topi kelin
ci maret sibuk jamu teh si kura palsu cerita kisah hidup banyak',
 'aku tahu sekarang lebih banyak luka hati bapak banding tubuh mamak lebih banyak
tangis hati mamak banding mata buah kisah jalan pulang lalu tarung tarung peluk er
at semua benci rasa sakit novel naskah awal asli tulis',
 'aku jadi tenang cinta banyak khawatir',
 'tiap orang berapa usia wujud impi jadi nyata semua orang hasil laku banyak fakto
r halang gagal usaha gapai sebut salah satu jelas visi visi mutlak milik orang wuj
ud impi benar visi visi bantu wujud impi bagaimana milik visi sesuai hendak tuhan
tahu lebih jauh sebut al about vision karya pdm nelson adu wajib baca beri paham v
isi buku saksi beberapa orang hasil wujud impi biar erti visi tinggal hati hidup v
isi tuhan gapai impi izin sebut kerja tuntun langkah hidup tuju masa depan penuh h
arap',
 'jangan sembarang terima beri nasihat orang tua dulu kau telanjur minta paket had
iah sekaligus kutu iblis kasih beri pasang sepatu merah kau kutuk tualang lebih te
pat gentayang naung tak rumah buah novel format pilih sendiri tualang gentayang ki
sah jalan cerabut potret goda batas gerak sangkut kabur tangkap gantung jalan mana
kau pilih tualang kutuk sepatu merah bawa new york kota tikus batas tijuana gereja
haarlem masjid jakarta taksi pengap kereta tak mau henti hidup mati bosan lama gen
tayang kau temu cerita kelana turis migran lari seberang cari atas rumah rute pint
u darurat cewek baik masuk surga cewek bandel gentayang',
 'who a breathing meat why some breathing meat lucky enough to understand their pu
rpose of existence but most of the meats have no idea why they exist in the univer
se the meat who knows and the meat who doesn e et know experience the same pain un
certainty and heartbreak but have different points of view on everything where und
er the sky on the soil between the air and the sea in the arms of people who treat
you badly what something you don e et know and something you know if you e ere luc
ky you e ell find the answer but other questions will come to you soon after that
when not today',
 'tahun 2006 amba pergi pulau buru cari orang kasih beri orang anak luar nikah lak
i bhisma dokter lulus leipzig jerman timur hilang tangkap perintah orde baru buang
pulau buru kamp tahan politik bubar tapol pulang bhisma tetap tak novel latar seja
rah kisah cinta hidup amba anak orang guru buah kota kecil jawa tengah e eaku besa
r kadipura aku tumbuh keluarga baca kitab tua e e tinggal kota diri temu bhisma ci

```
nta putus tiba peristiwa g30s yogyakarta buah serbu bhisma hilang lama baru pulau
buru amba tahu bhisma tak',
 'aku memang orang sahabat selalu kau butuh sahabat selalu pinjam pundak sandar ka
u sedang sedih kau lihat pandang beda lihat aku bagai orang lelaki cinta bukan bag
ai sahabat mungkin semua bukan salah aku salah takut ungkap maaf aku aku sakit']
```

In [ ]:
```
# Mengganti nilai-nilai NaN dengan string kosong
test_corpus.fillna('', inplace=True)

# Memanggil fungsi normalize_corpus setelah menangani nilai-nilai yang kosong
norm_test_corpus = normalize_corpus(test_corpus)
```

In [ ]:
```
test_corpus.fillna('', inplace=True)

norm_test_corpus[0:10]
```

Out[ ]: ['kisah unik gaul manusia antarkelas sosial bagai karakter tak kalah tarik nick da
tang baru new york percaya orang simpan rahasia daisy meski mulut pedas taksir ban
yak laki balik selalu usaha rupa tutup sakit hati khianat suami tom suami daisy se
kaligus teman kuliah nick lelaki didik tinggi kaya perangai kasar suka selingkuh t
ak suka selingkuh jordan perempuan muda cantik sinis pacar nick nick tahu selingku
h laku daisy tom pasang masingmasing terbit terbit serambi ilmu semesta serambi gr
oup',
 'james herriot baru lulus bagai dokter hewan tiba desa kecil darrowby yorkshire t
ak punya bayang sedikit hidup jalan teman baru jumpa bagai tualang tunggu buku cer
ita tahun pertama james herriot bagai dokter hewan desa darrowby indah hidup hari
sama siegfried farnon eksentrik tristan farnon selalu sial ternak binatang andal b
antu gaya cerita khas buat cinta seluruh dunia james herriot tutur cerita penuh ta
wa suka cita manusia binatang alam desa indah e edia cerita da',
 'gawat thor hilang benda paling harga milik apa kalau bukan mjolnirpalu thoryang
rupa senjata paling kuat sembilan dunia parah kali mjolnir jatuh tangan thrym si r
aja jotun sembunyi suatu tempat rahasia mjolnir musuh asgard manfaat situasi kerah
serang akibat mungkin picu jadi ragnarokpeperangan tanda akhir dunia mimpi magnus
datang loki aku rancang janji mjolnir syarat magnus bawa calon pengantin perempuan
maskawin sembah thrym magnus kawan tak punya pilih mesti segera jalan misi mjolnir
sekalipun penuh semua syarat waktu singkat bukan mudah mizan noura books novel fan
tasi terjemah percy thor asgard indonesia',
 'economic development in south sumatera',
 'eun ha won benar cinderella sungguh tak sangka kakek ha won jalan nyata direktur
kang epengusaha korea sangat kaya sedia bantu wujud ha won keluar rumah tinggal sa
ma sang ibu tiri ha won kemudian tinggal rumah mewah sama tiga cucu tampan sang di
rektur awal pribadi keren jadi kaya tak arti hidup ha won jadi mulus banyak iri de
kat kakakberadik keluarga kang tambah entah kang ji woon esang cucu tiga emembenci
nya sejak temu pertama kapan tenang',
 'tiap anjing milik punya kisah unik masingmasing e james herriot e es dog stories
kumpul kisah sejati anjing milik jadi hari james herriot dokter hewan desa yorkshi
re indah bagai kisah buku buat tertawa menang kisah gyp anjing gembala gonggong sa
tu kali hidup',
 'buck anjing lahir besar rumah hakim miller nyaman hangat california suatu hari b
uck culik jual jadi anjing tarik kereta daerah yukon selimut salju iklim ganas hid
up keras buat hari ubah drastis alam liar laku hukum taring pukul siapa kuat tahan
sini buck bukti tangguh semangat tak mudah patah e pertama kali terbit tahun 1903
the call of the wild rupa karya masterpiece jack london dasar alam bagai cari emas
alam liar canada the call of the wild cerita semangat tak kenal serah juang tahan
hidup klondike alaska',
 'timur antologi karya tulis indonesia timur hasil kuras tim makassar internationa
l writers festival miwf edisi perdana hadir puisi cerita pendek sebelas tulis pern
ah pilih jadi tulis undang miwf sejak 2011 sajak cerita pendek ama achmad cicilia
oday deasy tirayoh dicky senda emil amir erni aladjai faisal oddang ibe s palogai
irma agryanti jamil massa mario f lawi bawa makin kenal daerah indonesia timur mul
ai identitas adat budaya jalan hingga isu sosial beda tiap daerah selamat jajah in
donesia timur',
 'history of starvation and communism',
 'tragedi malam benar mimpi buruk orang manusia api urung ayah carter sadie kane p
eti mati tenggelam bawah lapis bumi kakak adik jebak buah tualang tegang uak rahas
ia keluarga hilang sang ayah sengaja bangkit lima dewa mesir kuno kini salah satu
dewa mesir suka buat onar set mengicar nyawa carter sadie bekal sedikit tahu kuat
magis terus juang selamat diri cari ayah set incar nyawa kakak adik mampu temu aya
h e epenggabungan mitologi dunia modern cara genius e e e esusan carpenter los ang
eles times e e tualang fantasi cerita sangat suka gemar seri percy jackson and the
olympians kejut akhir cerita buat pembac']

In [ ]:

In [ ]:

In [ ]:

Ekstraksi Fitur

Silakan isi bagian ini dengan script untuk keperluan ekstraksi fitur data teks yang akan digunakan untuk pengembangan model klasifikasi teks. Untuk setiap strategi ekstraksi fitur yang akan digunakan sebaiknya dibuat fungsi tersendiri dan dipanggil dalam satu fungsi ekstraksi fitur. Jangan lupa pada bagian ini ditampilkan contoh hasil ekstraksi fitur terhadap datanya.

In [ ]:
```
# Fungsi untuk mengekstraksi feature menggunakan TF-IDF Model
def tfidf_transformer(bow_matrix):
    transformer = TfidfTransformer(norm='l2', smooth_idf=True, use_idf=True)
    tfidf_matrix = transformer.fit_transform(bow_matrix)
    return transformer, tfidf_matrix

def tfidf_extractor(corpus, ngram_range=(1,1)):
    vectorizer = TfidfVectorizer(min_df=1, norm='l2', smooth_idf=True, use_idf=True
    features = vectorizer.fit_transform(corpus)
    return vectorizer, features
```

In [ ]:
```
 #mengunakan model TF-IDF untuk mengekstraksi feature
tfidf_vectorizer, tfidf_train_features = tfidf_extractor(norm_train_corpus)

# Menggunakan model TF-IDF yang sama untuk mengekstraksi feature dari test data
tfidf_test_features = tfidf_vectorizer.transform(norm_test_corpus)
```

In [ ]:
```
train_features=tfidf_train_features
train_labels=train_labels
test_features=tfidf_test_features
test_labels=test_labels
```

In [ ]:
```
df_train = pd.DataFrame()
df_train['train_feature'] = norm_train_corpus
df_train['train_labels'] = train_labels

df_test = pd.DataFrame()
df_test['test_feature'] = norm_test_corpus
df_test['test_labels'] = test_labels
```

In [ ]:
```
df_train.fillna('', inplace=True)

df_train.head()
```

Out[ ]:

	train_feature	train_labels
0	puji syukur panjat hadir allah swt atas ken te...	Education
1	buku fenomenal a brief history of time stephen...	Fiction
2	e edi arah sana e e si kucing lambai cakar kan...	Education
3	aku tahu sekarang lebih banyak luka hati bapak...	
4	aku jadi tenang cinta banyak khawatir	Education

In [ ]:
```
df_test.fillna('', inplace=True)

df_test.head()
```

Out[ ]:

	test_feature	test_labels
0	kisah unik gaul manusia antarkelas sosial baga...	
1	james herriot baru lulus bagai dokter hewan ti...	
2	gawat thor hilang benda paling harga milik apa...	
3	economic development in south sumatera	Economics
4	eun ha won benar cinderella sungguh tak sangka...	

Model Training

Silakan isi bagian ini dengan script untuk keperluan training model klasifikasi teks. Jangan lupa untuk menjabarkan rancangan pemodelan algoritme yang akan diterapkan dan kombinasi parameter yang akan digunakan dengan benar dan lengkap.

In [ ]:
```
# Convert the document text to a dense matrix
#train_features_dense = train_features.toarray()

# Convert the dense matrix to a list of strings
#train_features_list = [' '.join(map(str, row)) for row in train_features_dense]
```

In [ ]:
```
#SGDC Classifier

pipeline_SGDC = Pipeline([
    ('feature_selection', SelectFromModel(LinearSVC())),
    ('classification', SGDClassifier())
])


pipeline_LinearSVC = Pipeline([
    ('feature_selection', SelectFromModel(LinearSVC())),
    ('classification', LinearSVC())
])
```

```
pipeline_NB = Pipeline([
    ('feature_selection', SelectFromModel(LinearSVC())),
    ('classification', MultinomialNB())
])
```

In [ ]:
```
# Parameter yang akan
parameters_SGDC = {
      # Example parameter for TfidfVectorizer
    'feature_selection__estimator__C': [0.1, 1, 10],  # Example parameter for Linea
    'feature_selection__threshold': ['mean', 'median', '1.5*mean'],  # Example para
    'classification__loss' : ['hinge','perceptron'],  # Example parameter for SGDCl
    'classification__alpha': [0.0001, 0.001, 0.01],  # Example parameter for SGDCla
    'classification__penalty': ['l1', 'l2', 'elasticnet'],  # Example parameter for
    'classification__dual': ['auto'],# Example parameter for SGDClassifier
    'classification__max_iter' : [50,100],# Example parameter for SGDClassifier
    'classification__verbose' : [0,1,2,3,4,5,6,7,8,9,10],
    'classification__epsilon' : [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0],

}

parameters_LinearSVC = {
    'feature_selection__estimator__C': [0.1, 1, 10],  # Example parameter for Linea
    'feature_selection__threshold': ['mean', 'median', '1.5*mean'],  # Example para
    'classification__C': [0.1, 1, 10],  # Example parameter for LinearSVC
    'classification__loss': ['hinge', 'squared_hinge'],  # Example parameter for Li

    'classification__verbose': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10],  # Example param
    'classification__penalty': ['l1', 'l2'],  # Example parameter for LinearSVC
    'classification__max_iter': [50, 100, 1000],  # Example parameter for LinearSVC
    'classification__tol': [0.0001, 0.001, 0.01],  # Example parameter for LinearSV
}

parameters_NB = {
    'feature_selection__estimator__C': [0.1, 1, 10],  # Example parameter for Linea
    'feature_selection__threshold': ['mean', 'median', '1.5*mean'],  # Example para
    'classification__alpha': [0.0001, 0.001, 0.01],  # Example parameter for Multin
    'classification__fit_prior': [True, False],  # Example parameter for Multinomia

}
```

In [ ]:
```
# Inisialisasi objek GridSearchCV
grid_search_SGDC = GridSearchCV(pipeline_SGDC, parameters_SGDC, n_jobs=-1)

grid_search_LinearSVC = GridSearchCV(pipeline_LinearSVC, parameters_LinearSVC, n_jo

grid_search_NB = GridSearchCV(pipeline_NB, parameters_NB, n_jobs=-1)
```

In [ ]:
```
# Fit the model using the list of strings
grid_search_SGDC.fit(train_features, train_labels)
```

```
c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validation.py:42
5: FitFailedWarning:
89100 fits failed out of a total of 267300.
The score on these train-test partitions for these parameters will be set to nan.
If these failures are not expected, you can try to debug them by setting error_score
='raise'.

Below are more details about the failures:
--------------------------------------------------------------------------------
5639 fits failed with the following error:
Traceback (most recent call last):
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validati
on.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\pipeline.py", line 420, i
n fit
    self._final_estimator.fit(Xt, y, **fit_params_last_step)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1144, in w
rapper
    estimator._validate_params()
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in _v
alidate_params
    validate_parameter_constraints(
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.p
y", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'loss' parameter of SGDCl
assifier must be a str among {'huber', 'squared_hinge', 'perceptron', 'hinge', 'squa
red_epsilon_insensitive', 'modified_huber', 'epsilon_insensitive', 'log_loss', 'squa
red_error'}. Got 'log' instead.

--------------------------------------------------------------------------------
5656 fits failed with the following error:
Traceback (most recent call last):
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validati
on.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\pipeline.py", line 420, i
n fit
    self._final_estimator.fit(Xt, y, **fit_params_last_step)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1144, in w
rapper
    estimator._validate_params()
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in _v
alidate_params
    validate_parameter_constraints(
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.p
y", line 95, in validate_parameter_constraints
```

```
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'loss' parameter of SGDCl
assifier must be a str among {'modified_huber', 'perceptron', 'huber', 'hinge', 'squ
ared_hinge', 'log_loss', 'squared_error', 'squared_epsilon_insensitive', 'epsilon_in
sensitive'}. Got 'log' instead.

--------------------------------------------------------------------------------
5604 fits failed with the following error:
Traceback (most recent call last):
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validati
on.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\pipeline.py", line 420, i
n fit
    self._final_estimator.fit(Xt, y, **fit_params_last_step)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1144, in w
rapper
    estimator._validate_params()
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in _v
alidate_params
    validate_parameter_constraints(
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.p
y", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'loss' parameter of SGDCl
assifier must be a str among {'squared_error', 'squared_hinge', 'hinge', 'huber', 'l
og_loss', 'perceptron', 'modified_huber', 'squared_epsilon_insensitive', 'epsilon_in
sensitive'}. Got 'log' instead.

--------------------------------------------------------------------------------
5447 fits failed with the following error:
Traceback (most recent call last):
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validati
on.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\pipeline.py", line 420, i
n fit
    self._final_estimator.fit(Xt, y, **fit_params_last_step)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1144, in w
rapper
    estimator._validate_params()
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in _v
alidate_params
    validate_parameter_constraints(
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.p
y", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'loss' parameter of SGDCl
```

```
assifier must be a str among {'epsilon_insensitive', 'log_loss', 'huber', 'perceptro
n', 'squared_hinge', 'hinge', 'squared_epsilon_insensitive', 'modified_huber', 'squa
red_error'}. Got 'log' instead.

--------------------------------------------------------------------------------
5552 fits failed with the following error:
Traceback (most recent call last):
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validati
on.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\pipeline.py", line 420, i
n fit
    self._final_estimator.fit(Xt, y, **fit_params_last_step)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1144, in w
rapper
    estimator._validate_params()
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in _v
alidate_params
    validate_parameter_constraints(
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.p
y", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'loss' parameter of SGDCl
assifier must be a str among {'huber', 'perceptron', 'squared_error', 'squared_epsil
on_insensitive', 'modified_huber', 'log_loss', 'hinge', 'epsilon_insensitive', 'squa
red_hinge'}. Got 'log' instead.

--------------------------------------------------------------------------------
5501 fits failed with the following error:
Traceback (most recent call last):
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validati
on.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\pipeline.py", line 420, i
n fit
    self._final_estimator.fit(Xt, y, **fit_params_last_step)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1144, in w
rapper
    estimator._validate_params()
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in _v
alidate_params
    validate_parameter_constraints(
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.p
y", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'loss' parameter of SGDCl
assifier must be a str among {'squared_hinge', 'epsilon_insensitive', 'squared_epsil
on_insensitive', 'squared_error', 'hinge', 'log_loss', 'modified_huber', 'huber', 'p
```

```
erceptron'}. Got 'log' instead.

--------------------------------------------------------------------------------
5605 fits failed with the following error:
Traceback (most recent call last):
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validati
on.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\pipeline.py", line 420, i
n fit
    self._final_estimator.fit(Xt, y, **fit_params_last_step)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1144, in w
rapper
    estimator._validate_params()
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in _v
alidate_params
    validate_parameter_constraints(
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.p
y", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'loss' parameter of SGDCl
assifier must be a str among {'huber', 'squared_hinge', 'perceptron', 'modified_hube
r', 'squared_error', 'squared_epsilon_insensitive', 'hinge', 'log_loss', 'epsilon_in
sensitive'}. Got 'log' instead.

--------------------------------------------------------------------------------
5593 fits failed with the following error:
Traceback (most recent call last):
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validati
on.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\pipeline.py", line 420, i
n fit
    self._final_estimator.fit(Xt, y, **fit_params_last_step)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1144, in w
rapper
    estimator._validate_params()
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in _v
alidate_params
    validate_parameter_constraints(
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.p
y", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'loss' parameter of SGDCl
assifier must be a str among {'log_loss', 'epsilon_insensitive', 'squared_error', 'p
erceptron', 'squared_epsilon_insensitive', 'huber', 'squared_hinge', 'hinge', 'modif
ied_huber'}. Got 'log' instead.
```

```
--------------------------------------------------------------------------------
5601 fits failed with the following error:
Traceback (most recent call last):
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validati
on.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\pipeline.py", line 420, i
n fit
    self._final_estimator.fit(Xt, y, **fit_params_last_step)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1144, in w
rapper
    estimator._validate_params()
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in _v
alidate_params
    validate_parameter_constraints(
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.p
y", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'loss' parameter of SGDCl
assifier must be a str among {'perceptron', 'log_loss', 'huber', 'hinge', 'squared_e
rror', 'epsilon_insensitive', 'squared_hinge', 'squared_epsilon_insensitive', 'modif
ied_huber'}. Got 'log' instead.

--------------------------------------------------------------------------------
5626 fits failed with the following error:
Traceback (most recent call last):
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validati
on.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\pipeline.py", line 420, i
n fit
    self._final_estimator.fit(Xt, y, **fit_params_last_step)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1144, in w
rapper
    estimator._validate_params()
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in _v
alidate_params
    validate_parameter_constraints(
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.p
y", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'loss' parameter of SGDCl
assifier must be a str among {'squared_error', 'huber', 'squared_hinge', 'perceptro
n', 'log_loss', 'modified_huber', 'hinge', 'epsilon_insensitive', 'squared_epsilon_i
nsensitive'}. Got 'log' instead.

--------------------------------------------------------------------------------
5596 fits failed with the following error:
```

```
Traceback (most recent call last):
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validati
on.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\pipeline.py", line 420, i
n fit
    self._final_estimator.fit(Xt, y, **fit_params_last_step)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1144, in w
rapper
    estimator._validate_params()
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in _v
alidate_params
    validate_parameter_constraints(
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.p
y", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'loss' parameter of SGDCl
assifier must be a str among {'modified_huber', 'epsilon_insensitive', 'perceptron',
'hinge', 'log_loss', 'huber', 'squared_hinge', 'squared_error', 'squared_epsilon_ins
ensitive'}. Got 'log' instead.

--------------------------------------------------------------------------------
5544 fits failed with the following error:
Traceback (most recent call last):
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validati
on.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\pipeline.py", line 420, i
n fit
    self._final_estimator.fit(Xt, y, **fit_params_last_step)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1144, in w
rapper
    estimator._validate_params()
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in _v
alidate_params
    validate_parameter_constraints(
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.p
y", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'loss' parameter of SGDCl
assifier must be a str among {'squared_hinge', 'squared_epsilon_insensitive', 'perce
ptron', 'squared_error', 'hinge', 'huber', 'epsilon_insensitive', 'log_loss', 'modif
ied_huber'}. Got 'log' instead.

--------------------------------------------------------------------------------
5541 fits failed with the following error:
Traceback (most recent call last):
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validati
```

```
on.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\pipeline.py", line 420, i
n fit
    self._final_estimator.fit(Xt, y, **fit_params_last_step)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1144, in w
rapper
    estimator._validate_params()
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in _v
alidate_params
    validate_parameter_constraints(
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.p
y", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'loss' parameter of SGDCl
assifier must be a str among {'huber', 'epsilon_insensitive', 'modified_huber', 'log
_loss', 'squared_hinge', 'squared_error', 'squared_epsilon_insensitive', 'hinge', 'p
erceptron'}. Got 'log' instead.

--------------------------------------------------------------------------------
5449 fits failed with the following error:
Traceback (most recent call last):
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validati
on.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\pipeline.py", line 420, i
n fit
    self._final_estimator.fit(Xt, y, **fit_params_last_step)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1144, in w
rapper
    estimator._validate_params()
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in _v
alidate_params
    validate_parameter_constraints(
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.p
y", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'loss' parameter of SGDCl
assifier must be a str among {'hinge', 'huber', 'perceptron', 'log_loss', 'squared_h
inge', 'squared_error', 'squared_epsilon_insensitive', 'modified_huber', 'epsilon_in
sensitive'}. Got 'log' instead.

--------------------------------------------------------------------------------
5571 fits failed with the following error:
Traceback (most recent call last):
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validati
on.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
```

```
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\pipeline.py", line 420, i
n fit
    self._final_estimator.fit(Xt, y, **fit_params_last_step)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1144, in w
rapper
    estimator._validate_params()
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in _v
alidate_params
    validate_parameter_constraints(
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.p
y", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'loss' parameter of SGDCl
assifier must be a str among {'perceptron', 'epsilon_insensitive', 'squared_epsilon_
insensitive', 'squared_hinge', 'log_loss', 'squared_error', 'hinge', 'modified_hube
r', 'huber'}. Got 'log' instead.

--------------------------------------------------------------------------------
5575 fits failed with the following error:
Traceback (most recent call last):
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validati
on.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\pipeline.py", line 420, i
n fit
    self._final_estimator.fit(Xt, y, **fit_params_last_step)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1144, in w
rapper
    estimator._validate_params()
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in _v
alidate_params
    validate_parameter_constraints(
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.p
y", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'loss' parameter of SGDCl
assifier must be a str among {'squared_error', 'squared_hinge', 'log_loss', 'hinge',
'perceptron', 'modified_huber', 'epsilon_insensitive', 'huber', 'squared_epsilon_ins
ensitive'}. Got 'log' instead.

  warnings.warn(some_fits_failed_message, FitFailedWarning)
```

```
-- Epoch 1
Norm: 50.10, NNZs: 740, Bias: -0.888773, T: 226, Avg. loss: 0.218616
Total training time: 0.00 seconds.
-- Epoch 2
Norm: 46.44, NNZs: 785, Bias: -1.274279, T: 452, Avg. loss: 0.031459
Total training time: 0.00 seconds.
-- Epoch 3
Norm: 41.07, NNZs: 782, Bias: -1.274279, T: 678, Avg. loss: 0.000000
Total training time: 0.00 seconds.
-- Epoch 4
Norm: 37.78, NNZs: 805, Bias: -1.272864, T: 904, Avg. loss: 0.001752
Total training time: 0.00 seconds.
-- Epoch 5
Norm: 34.59, NNZs: 828, Bias: -1.320258, T: 1130, Avg. loss: 0.000410
Total training time: 0.00 seconds.
-- Epoch 6
Norm: 31.74, NNZs: 824, Bias: -1.320258, T: 1356, Avg. loss: 0.000000
Total training time: 0.00 seconds.
-- Epoch 7
Norm: 29.37, NNZs: 822, Bias: -1.320258, T: 1582, Avg. loss: 0.000000
Total training time: 0.00 seconds.
-- Epoch 8
Norm: 28.35, NNZs: 819, Bias: -1.246617, T: 1808, Avg. loss: 0.000897
Total training time: 0.00 seconds.
Convergence after 8 epochs took 0.00 seconds
-- Epoch 1
Norm: 57.25, NNZs: 1103, Bias: -1.045750, T: 226, Avg. loss: 0.356750
Total training time: 0.00 seconds.
-- Epoch 2
Norm: 50.75, NNZs: 1113, Bias: -1.279160, T: 452, Avg. loss: 0.002508
Total training time: 0.00 seconds.
-- Epoch 3
Norm: 44.88, NNZs: 1108, Bias: -1.279160, T: 678, Avg. loss: 0.000000
Total training time: 0.00 seconds.
-- Epoch 4
Norm: 40.30, NNZs: 1101, Bias: -1.279160, T: 904, Avg. loss: 0.000000
Total training time: 0.00 seconds.
-- Epoch 5
Norm: 37.71, NNZs: 1101, Bias: -1.179954, T: 1130, Avg. loss: 0.000611
Total training time: 0.00 seconds.
-- Epoch 6
Norm: 35.50, NNZs: 1108, Bias: -1.227431, T: 1356, Avg. loss: 0.001164
Total training time: 0.00 seconds.
-- Epoch 7
Norm: 33.83, NNZs: 1110, Bias: -1.187728, T: 1582, Avg. loss: 0.006228
Total training time: 0.00 seconds.
-- Epoch 8
Norm: 31.66, NNZs: 1107, Bias: -1.225521, T: 1808, Avg. loss: 0.000870
Total training time: 0.00 seconds.
Convergence after 8 epochs took 0.00 seconds
-- Epoch 1
Norm: 73.99, NNZs: 1393, Bias: -0.300654, T: 226, Avg. loss: 0.492358
Total training time: 0.00 seconds.
-- Epoch 2
Norm: 65.32, NNZs: 1409, Bias: -0.371425, T: 452, Avg. loss: 0.002927
Total training time: 0.00 seconds.
```

```
-- Epoch 3
Norm: 58.46, NNZs: 1418, Bias: -0.367585, T: 678, Avg. loss: 0.001829
Total training time: 0.00 seconds.
-- Epoch 4
Norm: 53.11, NNZs: 1414, Bias: -0.257672, T: 904, Avg. loss: 0.001381
Total training time: 0.00 seconds.
-- Epoch 5
Norm: 48.87, NNZs: 1421, Bias: -0.258445, T: 1130, Avg. loss: 0.000516
Total training time: 0.00 seconds.
-- Epoch 6
Norm: 45.85, NNZs: 1450, Bias: -0.257540, T: 1356, Avg. loss: 0.002165
Total training time: 0.00 seconds.
-- Epoch 7
Norm: 43.11, NNZs: 1481, Bias: -0.297206, T: 1582, Avg. loss: 0.001273
Total training time: 0.00 seconds.
-- Epoch 8
Norm: 41.45, NNZs: 1530, Bias: -0.223233, T: 1808, Avg. loss: 0.002878
Total training time: 0.00 seconds.
Convergence after 8 epochs took 0.00 seconds
-- Epoch 1
Norm: 43.86, NNZs: 693, Bias: -1.065525, T: 226, Avg. loss: 0.152586
Total training time: 0.00 seconds.
-- Epoch 2
Norm: 38.92, NNZs: 699, Bias: -1.217715, T: 452, Avg. loss: 0.008028
Total training time: 0.00 seconds.
-- Epoch 3
Norm: 35.18, NNZs: 712, Bias: -1.157255, T: 678, Avg. loss: 0.001940
Total training time: 0.00 seconds.
-- Epoch 4
Norm: 31.80, NNZs: 726, Bias: -1.215906, T: 904, Avg. loss: 0.004893
Total training time: 0.00 seconds.
-- Epoch 5
Norm: 29.62, NNZs: 724, Bias: -1.166031, T: 1130, Avg. loss: 0.000161
Total training time: 0.00 seconds.
-- Epoch 6
Norm: 27.48, NNZs: 726, Bias: -1.212629, T: 1356, Avg. loss: 0.000759
Total training time: 0.00 seconds.
-- Epoch 7
Norm: 25.42, NNZs: 719, Bias: -1.212629, T: 1582, Avg. loss: 0.000000
Total training time: 0.00 seconds.
-- Epoch 8
Norm: 23.67, NNZs: 715, Bias: -1.212629, T: 1808, Avg. loss: 0.000000
Total training time: 0.00 seconds.
-- Epoch 9
Norm: 22.69, NNZs: 710, Bias: -1.178616, T: 2034, Avg. loss: 0.000730
Total training time: 0.00 seconds.
-- Epoch 10
Norm: 21.86, NNZs: 703, Bias: -1.146074, T: 2260, Avg. loss: 0.000084
Total training time: 0.00 seconds.
Convergence after 10 epochs took 0.00 seconds
-- Epoch 1
Norm: 67.86, NNZs: 1289, Bias: -0.594475, T: 226, Avg. loss: 0.529200
Total training time: 0.00 seconds.
-- Epoch 2
Norm: 61.54, NNZs: 1363, Bias: -0.895273, T: 452, Avg. loss: 0.030520
Total training time: 0.00 seconds.
```

```
-- Epoch 3
Norm: 54.88, NNZs: 1353, Bias: -0.834886, T: 678, Avg. loss: 0.000349
Total training time: 0.00 seconds.
-- Epoch 4
Norm: 49.93, NNZs: 1347, Bias: -0.837419, T: 904, Avg. loss: 0.001315
Total training time: 0.00 seconds.
-- Epoch 5
Norm: 46.40, NNZs: 1345, Bias: -0.789297, T: 1130, Avg. loss: 0.001617
Total training time: 0.00 seconds.
-- Epoch 6
Norm: 43.20, NNZs: 1379, Bias: -0.925154, T: 1356, Avg. loss: 0.000554
Total training time: 0.00 seconds.
-- Epoch 7
Norm: 40.73, NNZs: 1379, Bias: -0.884224, T: 1582, Avg. loss: 0.004059
Total training time: 0.00 seconds.
-- Epoch 8
Norm: 38.69, NNZs: 1390, Bias: -0.847016, T: 1808, Avg. loss: 0.001113
Total training time: 0.00 seconds.
Convergence after 8 epochs took 0.00 seconds
-- Epoch 1
Norm: 50.77, NNZs: 743, Bias: -1.051774, T: 226, Avg. loss: 0.196964
Total training time: 0.00 seconds.
-- Epoch 2
Norm: 46.11, NNZs: 780, Bias: -1.192347, T: 452, Avg. loss: 0.005315
Total training time: 0.00 seconds.
-- Epoch 3
Norm: 41.30, NNZs: 776, Bias: -1.130808, T: 678, Avg. loss: 0.006911
Total training time: 0.00 seconds.
-- Epoch 4
Norm: 37.09, NNZs: 767, Bias: -1.130808, T: 904, Avg. loss: 0.000000
Total training time: 0.00 seconds.
-- Epoch 5
Norm: 33.71, NNZs: 766, Bias: -1.130808, T: 1130, Avg. loss: 0.000000
Total training time: 0.00 seconds.
-- Epoch 6
Norm: 30.94, NNZs: 763, Bias: -1.130808, T: 1356, Avg. loss: 0.000000
Total training time: 0.00 seconds.
-- Epoch 7
Norm: 29.89, NNZs: 782, Bias: -1.091468, T: 1582, Avg. loss: 0.001035
Total training time: 0.00 seconds.
-- Epoch 8
Norm: 28.48, NNZs: 802, Bias: -1.094016, T: 1808, Avg. loss: 0.001401
Total training time: 0.00 seconds.
-- Epoch 9
Norm: 27.25, NNZs: 813, Bias: -1.095551, T: 2034, Avg. loss: 0.001049
Total training time: 0.00 seconds.
Convergence after 9 epochs took 0.00 seconds
Best Score:  0.71256038647343
Best Parameters:  {'classification__alpha': 0.0001, 'classification__epsilon': 0.2,
'classification__loss': 'hinge', 'classification__max_iter': 100, 'classification__p
enalty': 'elasticnet', 'classification__verbose': 6, 'feature_selection__estimator__
C': 0.1, 'feature_selection__threshold': 'median'}
Best Estimator:  Pipeline(steps=[('feature_selection',
                 SelectFromModel(estimator=LinearSVC(C=0.1),
                                 threshold='median')),
                ('classification',
```

```
                    SGDClassifier(epsilon=0.2, max_iter=100, penalty='elasticnet',
                                  verbose=6))])
```

```
c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_search.py:976: U
serWarning: One or more of the test scores are non-finite: [0.64599034 0.65932367 0.
65043478 ... 0.4868599  0.47826087 0.48657005]
  warnings.warn(
c:\ProgramData\anaconda3\Lib\site-packages\sklearn\svm\_classes.py:32: FutureWarnin
g: The default value of `dual` will change from `True` to ``'auto'`` in 1.5. Set the v
alue of `dual` explicitly to suppress the warning.
  warnings.warn(
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    0.0s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   2 out of   2 | elapsed:    0.0s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   3 out of   3 | elapsed:    0.0s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   4 out of   4 | elapsed:    0.0s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   5 out of   5 | elapsed:    0.0s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   6 out of   6 | elapsed:    0.0s finished
```

In [ ]:
```
print("Best Score: ", grid_search_SGDC.best_score_)
print("Best Parameters: ", grid_search_SGDC.best_params_)
print("Best Estimator: ", grid_search_SGDC.best_estimator_)
```

```
Best Score:  0.71256038647343
Best Parameters:  {'classification__alpha': 0.0001, 'classification__epsilon': 0.2,
'classification__loss': 'hinge', 'classification__max_iter': 100, 'classification__p
enalty': 'elasticnet', 'classification__verbose': 6, 'feature_selection__estimator__
C': 0.1, 'feature_selection__threshold': 'median'}
Best Estimator:  Pipeline(steps=[('feature_selection',
                 SelectFromModel(estimator=LinearSVC(C=0.1),
                                 threshold='median')),
                ('classification',
                 SGDClassifier(epsilon=0.2, max_iter=100, penalty='elasticnet',
                               verbose=6))])
```

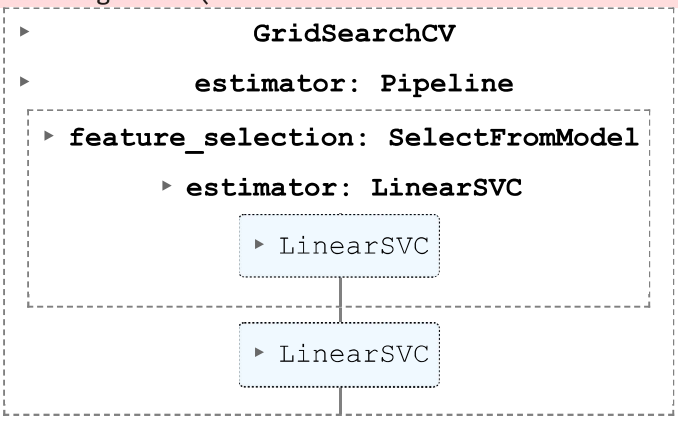
In [ ]:
```
grid_search_LinearSVC.fit(train_features, train_labels)
```

```
c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validation.py:42
5: FitFailedWarning:
35640 fits failed out of a total of 71280.
The score on these train-test partitions for these parameters will be set to nan.
If these failures are not expected, you can try to debug them by setting error_score
='raise'.

Below are more details about the failures:
--------------------------------------------------------------------------------
8910 fits failed with the following error:
Traceback (most recent call last):
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validati
on.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\pipeline.py", line 420, i
n fit
    self._final_estimator.fit(Xt, y, **fit_params_last_step)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\svm\_classes.py", line 31
5, in fit
    self.coef_, self.intercept_, n_iter_ = _fit_liblinear(
                                           ^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\svm\_base.py", line 1221,
in _fit_liblinear
    solver_type = _get_liblinear_solver_type(multi_class, penalty, loss, dual)
                  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\svm\_base.py", line 1060,
in _get_liblinear_solver_type
    raise ValueError(
ValueError: Unsupported set of arguments: The combination of penalty='l1' and loss
='hinge' is not supported, Parameters: penalty='l1', loss='hinge', dual=True

--------------------------------------------------------------------------------
8910 fits failed with the following error:
Traceback (most recent call last):
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validati
on.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\pipeline.py", line 420, i
n fit
    self._final_estimator.fit(Xt, y, **fit_params_last_step)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\svm\_classes.py", line 31
5, in fit
    self.coef_, self.intercept_, n_iter_ = _fit_liblinear(
                                           ^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\svm\_base.py", line 1221,
in _fit_liblinear
    solver_type = _get_liblinear_solver_type(multi_class, penalty, loss, dual)
                  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\svm\_base.py", line 1060,
in _get_liblinear_solver_type
    raise ValueError(
ValueError: Unsupported set of arguments: The combination of penalty='l1' and loss
='squared_hinge' are not supported when dual=True, Parameters: penalty='l1', loss='s
quared_hinge', dual=True

--------------------------------------------------------------------------------
8910 fits failed with the following error:
Traceback (most recent call last):
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validati
on.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\pipeline.py", line 420, i
n fit
    self._final_estimator.fit(Xt, y, **fit_params_last_step)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\svm\_classes.py", line 31
5, in fit
    self.coef_, self.intercept_, n_iter_ = _fit_liblinear(
                                           ^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\svm\_base.py", line 1221,
in _fit_liblinear
    solver_type = _get_liblinear_solver_type(multi_class, penalty, loss, dual)
                  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\svm\_base.py", line 1060,
in _get_liblinear_solver_type
    raise ValueError(
ValueError: Unsupported set of arguments: The combination of penalty='l1' and loss
='hinge' is not supported, Parameters: penalty='l1', loss='hinge', dual=False

--------------------------------------------------------------------------------
8910 fits failed with the following error:
Traceback (most recent call last):
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validati
on.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\pipeline.py", line 420, i
n fit
    self._final_estimator.fit(Xt, y, **fit_params_last_step)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\svm\_classes.py", line 31
5, in fit
    self.coef_, self.intercept_, n_iter_ = _fit_liblinear(
                                           ^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\svm\_base.py", line 1221,
in _fit_liblinear
    solver_type = _get_liblinear_solver_type(multi_class, penalty, loss, dual)
                  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\svm\_base.py", line 1060,
in _get_liblinear_solver_type
    raise ValueError(
ValueError: Unsupported set of arguments: The combination of penalty='l2' and loss
='hinge' are not supported when dual=False, Parameters: penalty='l2', loss='hinge',
dual=False

  warnings.warn(some_fits_failed_message, FitFailedWarning)
c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_search.py:976: U
serWarning: One or more of the test scores are non-finite: [       nan        nan
nan ... 0.64599034 0.64589372 0.63285024]
  warnings.warn(
c:\ProgramData\anaconda3\Lib\site-packages\sklearn\svm\_classes.py:32: FutureWarnin
g: The default value of `dual` will change from `True` to `'auto'` in 1.5. Set the v
alue of `dual` explicitly to suppress the warning.
  warnings.warn(
c:\ProgramData\anaconda3\Lib\site-packages\sklearn\svm\_base.py:1242: ConvergenceWar
ning: Liblinear failed to converge, increase the number of iterations.
  warnings.warn(
```

Out[ ]:



In [ ]:

```
print("Best Score: ", grid_search_LinearSVC.best_score_)
print("Best Parameters: ", grid_search_LinearSVC.best_params_)
print("Best Estimator: ", grid_search_LinearSVC.best_estimator_)
```

```
Best Score:  0.6593236714975845
Best Parameters:  {'classification__C': 10, 'classification__dual': True, 'classific
ation__loss': 'hinge', 'classification__max_iter': 50, 'classification__penalty': 'l
2', 'classification__tol': 0.0001, 'classification__verbose': 0, 'feature_selection_
_estimator__C': 10, 'feature_selection__threshold': 'mean'}
Best Estimator:  Pipeline(steps=[('feature_selection',
                 SelectFromModel(estimator=LinearSVC(C=10), threshold='mean')),
                ('classification',
                 LinearSVC(C=10, dual=True, loss='hinge', max_iter=50))])
```

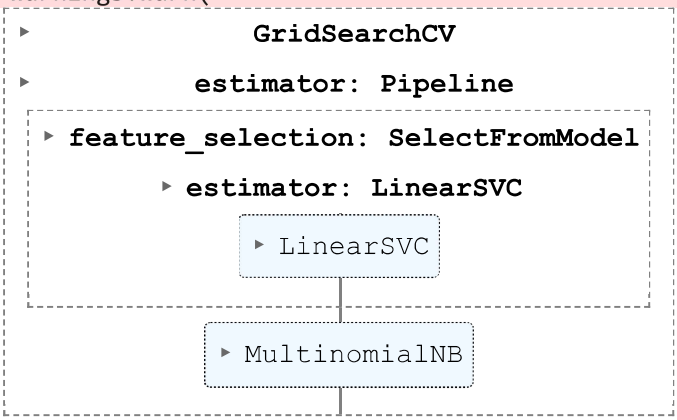
In [ ]:
```
grid_search_NB.fit(train_features, train_labels)
```

```
c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validation.py:42
5: FitFailedWarning:
2430 fits failed out of a total of 2700.
The score on these train-test partitions for these parameters will be set to nan.
If these failures are not expected, you can try to debug them by setting error_score
='raise'.

Below are more details about the failures:
--------------------------------------------------------------------------------
2430 fits failed with the following error:
Traceback (most recent call last):
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_validati
on.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\pipeline.py", line 420, i
n fit
    self._final_estimator.fit(Xt, y, **fit_params_last_step)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 1151, in w
rapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\naive_bayes.py", line 77
5, in fit
    self._update_class_log_prior(class_prior=class_prior)
  File "c:\ProgramData\anaconda3\Lib\site-packages\sklearn\naive_bayes.py", line 59
0, in _update_class_log_prior
    raise ValueError("Number of priors must match number of classes.")
ValueError: Number of priors must match number of classes.

  warnings.warn(some_fits_failed_message, FitFailedWarning)
c:\ProgramData\anaconda3\Lib\site-packages\sklearn\model_selection\_search.py:976: U
serWarning: One or more of the test scores are non-finite: [0.68589372 0.68154589 0.
66821256 0.68589372 0.69043478 0.68144928
 0.69033816 0.68599034 0.68144928 0.70376812 0.69033816 0.672657
 0.69487923 0.69922705 0.67710145 0.69487923 0.69033816 0.67710145
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
```

```
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
 0.68144928 0.67700483 0.67710145 0.68589372 0.68599034 0.67700483
 0.67700483 0.68154589 0.67700483 0.69932367 0.70811594 0.67710145
 0.69487923 0.70357488 0.68599034 0.69487923 0.70801932 0.68599034
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
 0.68589372 0.68154589 0.67256039 0.69033816 0.68608696 0.67700483
 0.68154589 0.68608696 0.66821256 0.70830918 0.72135266 0.69913043
 0.69932367 0.71690821 0.69487923 0.69932367 0.70357488 0.68154589
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
```

```
       nan        nan        nan        nan        nan        nan
       nan        nan        nan        nan        nan        nan
       nan        nan        nan        nan        nan        nan
       nan        nan        nan        nan        nan        nan
       nan        nan        nan        nan        nan        nan
       nan        nan        nan        nan        nan        nan
       nan        nan        nan        nan        nan        nan
       nan        nan        nan        nan        nan        nan
       nan        nan        nan        nan        nan        nan
       nan        nan        nan        nan        nan        nan
       nan        nan        nan        nan        nan        nan
       nan        nan        nan        nan        nan        nan
       nan        nan        nan        nan        nan        nan
       nan        nan        nan        nan        nan        nan]
  warnings.warn(
c:\ProgramData\anaconda3\Lib\site-packages\sklearn\svm\_classes.py:32: FutureWarnin
g: The default value of `dual` will change from `True` to ``'auto'`` in 1.5. Set the v
alue of `dual` explicitly to suppress the warning.
  warnings.warn(
```

Out[ ]:



In [ ]:
```
print("Best Score: ", grid_search_NB.best_score_)
print("Best Parameters: ", grid_search_NB.best_params_)
print("Best Estimator: ", grid_search_NB.best_estimator_)
```

```
Best Score:  0.7213526570048309
Best Parameters:  {'classification__alpha': 0.01, 'classification__class_prior': Non
e, 'classification__fit_prior': False, 'feature_selection__estimator__C': 0.1, 'feat
ure_selection__threshold': 'median'}
Best Estimator:  Pipeline(steps=[('feature_selection',
                 SelectFromModel(estimator=LinearSVC(C=0.1),
                                 threshold='median')),
                ('classification', MultinomialNB(alpha=0.01, fit_prior=False))])
```

Model Evaluation

Silakan isi bagian ini dengan script untuk keperluan evaluasi model klasifikasi teks. Jangan lupa untuk menjabarkan rancangan proses evaluasi model, metrik, dan visualisasi confusion matrix yang akan digunakan dengan benar dan lengkap

In [ ]:
```
def get_metrics(true_labels, predicted_labels):
    print('Accuracy: ', np.round(metrics.accuracy_score(true_labels, predicted_labe
    print('Precision: ', np.round(metrics.precision_score(true_labels, predicted_la
```

```
    print('Recall: ', np.round(metrics.recall_score(true_labels, predicted_labels,
    print('F1 Score: ', np.round(metrics.f1_score(true_labels, predicted_labels, av
```

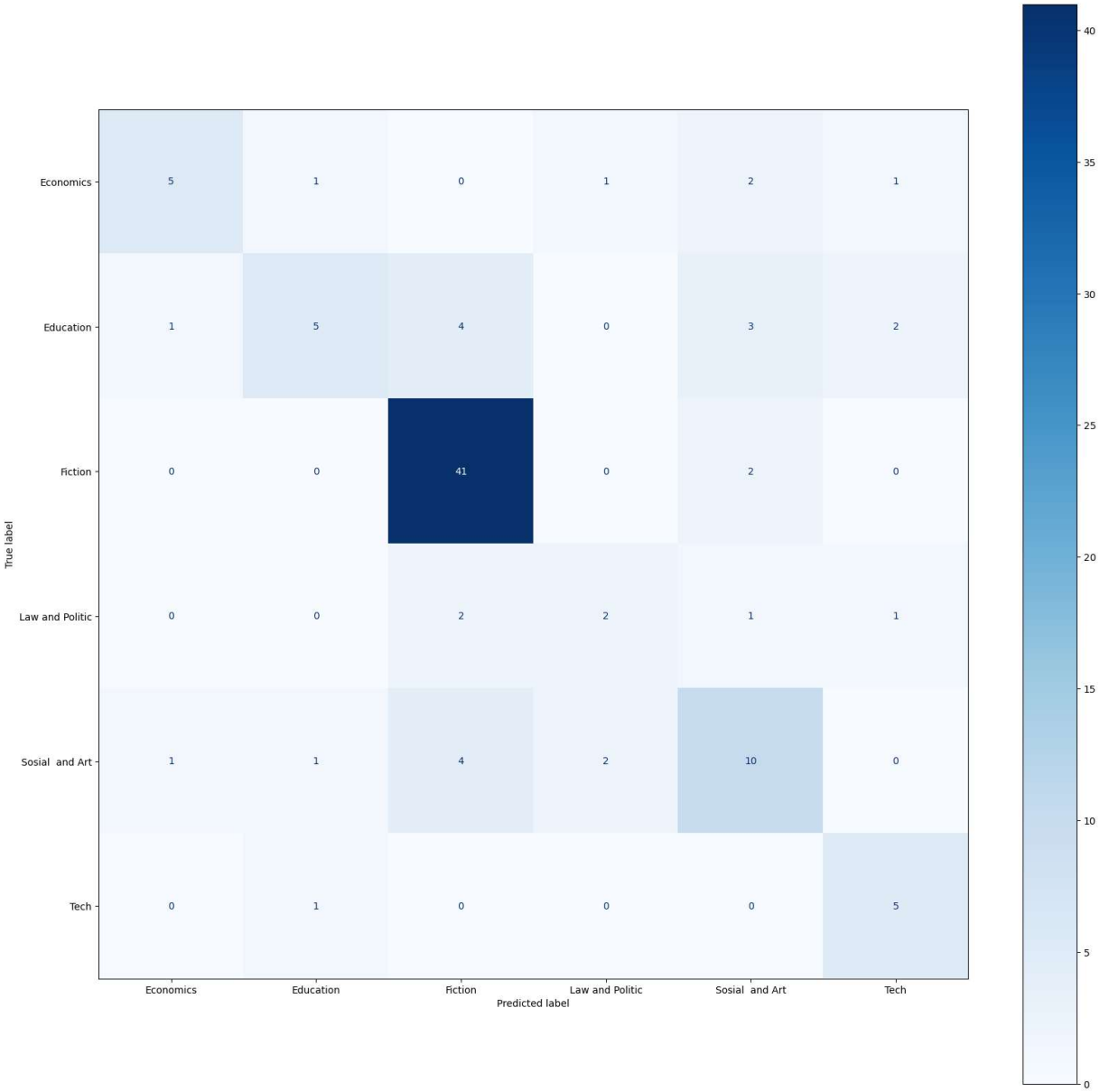
In [ ]:
```
def get_confusion_matrix(true_labels, predicted_labels):
    cm = metrics.confusion_matrix(y_true=true_labels, y_pred=predicted_labels)
    return cm
```

In [ ]:
```
# Prediksi menggunakan model Model



predictions_SGDC = grid_search_SGDC.best_estimator_.predict(test_features)

predictions_LinearSVC = grid_search_LinearSVC.best_estimator_.predict(test_features

predictions_NB = grid_search_NB.best_estimator_.predict(test_features)
```

In [ ]:
```
#Evaluasi Masing-Masing mode

print("SGDC Metrics:")

get_metrics(true_labels=test_labels, predicted_labels=predictions_SGDC)

print("Classification Report:")
print(metrics.classification_report(test_labels, predictions_SGDC))
print("Confusion Matrix:")
print(get_confusion_matrix(test_labels, predictions_SGDC))



disp = ConfusionMatrixDisplay(confusion_matrix=get_confusion_matrix(test_labels, pr

fig, ax = plt.subplots(figsize=(20, 20))


disp.plot(ax=ax, cmap=plt.cm.Blues)
```

```
SGDC Metrics:
Accuracy:  0.69
Precision:  0.68
Recall:  0.69
F1 Score:  0.67
Classification Report:
                 precision    recall  f1-score   support

      Economics       0.71      0.50      0.59        10
      Education       0.62      0.33      0.43        15
        Fiction       0.80      0.95      0.87        43
Law and Politic       0.40      0.33      0.36         6
Sosial  and Art       0.56      0.56      0.56        18
           Tech       0.56      0.83      0.67         6

       accuracy                           0.69        98
      macro avg       0.61      0.58      0.58        98
   weighted avg       0.68      0.69      0.67        98

Confusion Matrix:
[[ 5  1  0  1  2  1]
 [ 1  5  4  0  3  2]
 [ 0  0 41  0  2  0]
 [ 0  0  2  2  1  1]
 [ 1  1  4  2 10  0]
 [ 0  1  0  0  0  5]]
```

Out[ ]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x26851d8ad90>

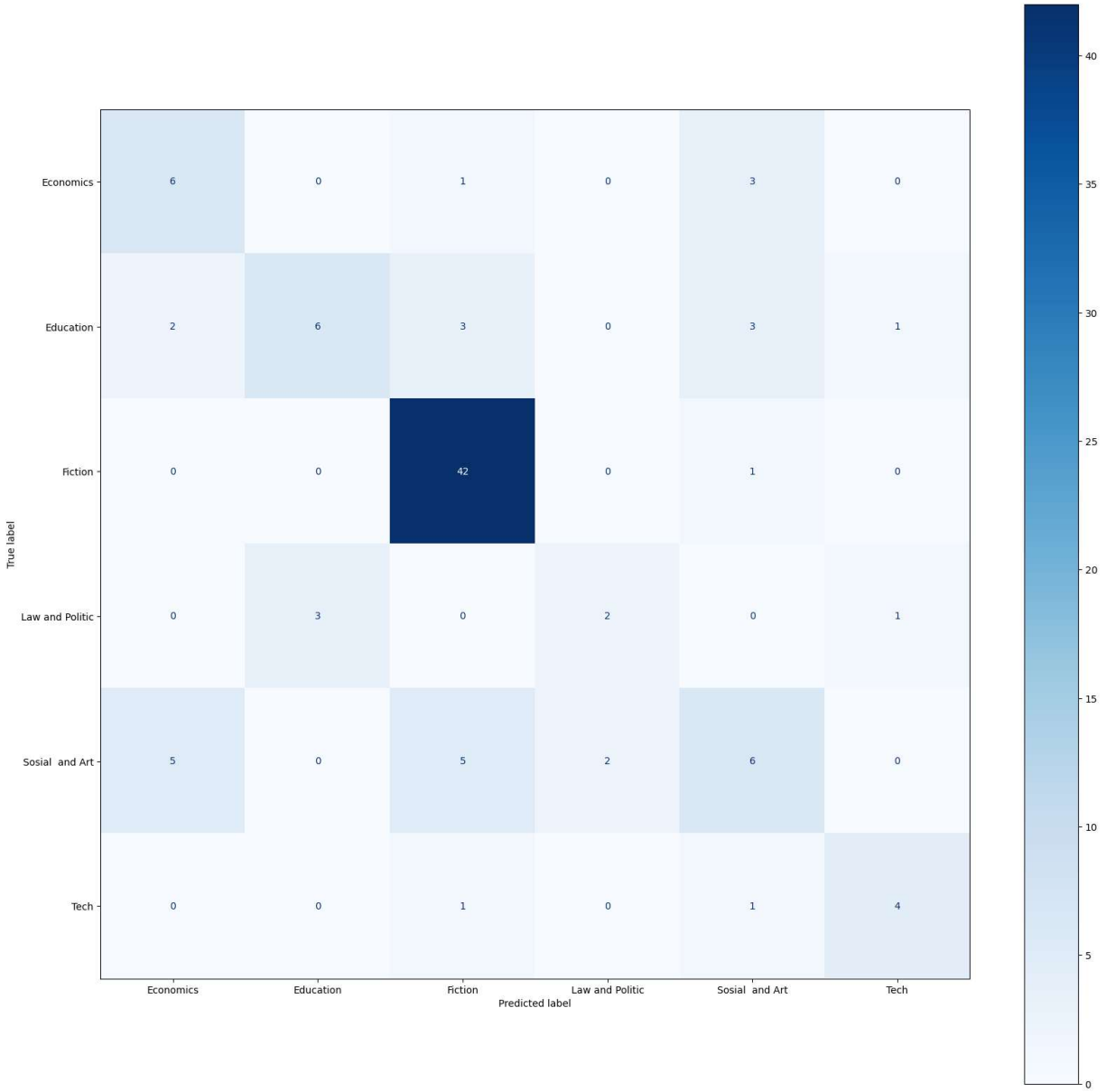


In [ ]:

```
print("LinearSVC Metrics:")

get_metrics(true_labels=test_labels, predicted_labels=predictions_LinearSVC)

print("Classification Report:")
print(metrics.classification_report(test_labels, predictions_LinearSVC))
print("Confusion Matrix:")
print(get_confusion_matrix(test_labels, predictions_LinearSVC))


disp = ConfusionMatrixDisplay(confusion_matrix=get_confusion_matrix(test_labels, pr

fig, ax = plt.subplots(figsize=(20, 20))


disp.plot(ax=ax, cmap=plt.cm.Blues)
```

```
LinearSVC Metrics:
Accuracy:  0.67
Precision:  0.65
Recall:  0.67
F1 Score:  0.65
Classification Report:
                 precision    recall  f1-score   support

      Economics       0.46      0.60      0.52        10
      Education       0.67      0.40      0.50        15
        Fiction       0.81      0.98      0.88        43
Law and Politic       0.50      0.33      0.40         6
Sosial  and Art       0.43      0.33      0.38        18
           Tech       0.67      0.67      0.67         6

       accuracy                           0.67        98
      macro avg       0.59      0.55      0.56        98
   weighted avg       0.65      0.67      0.65        98

Confusion Matrix:
[[ 6  0  1  0  3  0]
 [ 2  6  3  0  3  1]
 [ 0  0 42  0  1  0]
 [ 0  3  0  2  0  1]
 [ 5  0  5  2  6  0]
 [ 0  0  1  0  1  4]]
```

Out[ ]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x26851dc5f50>

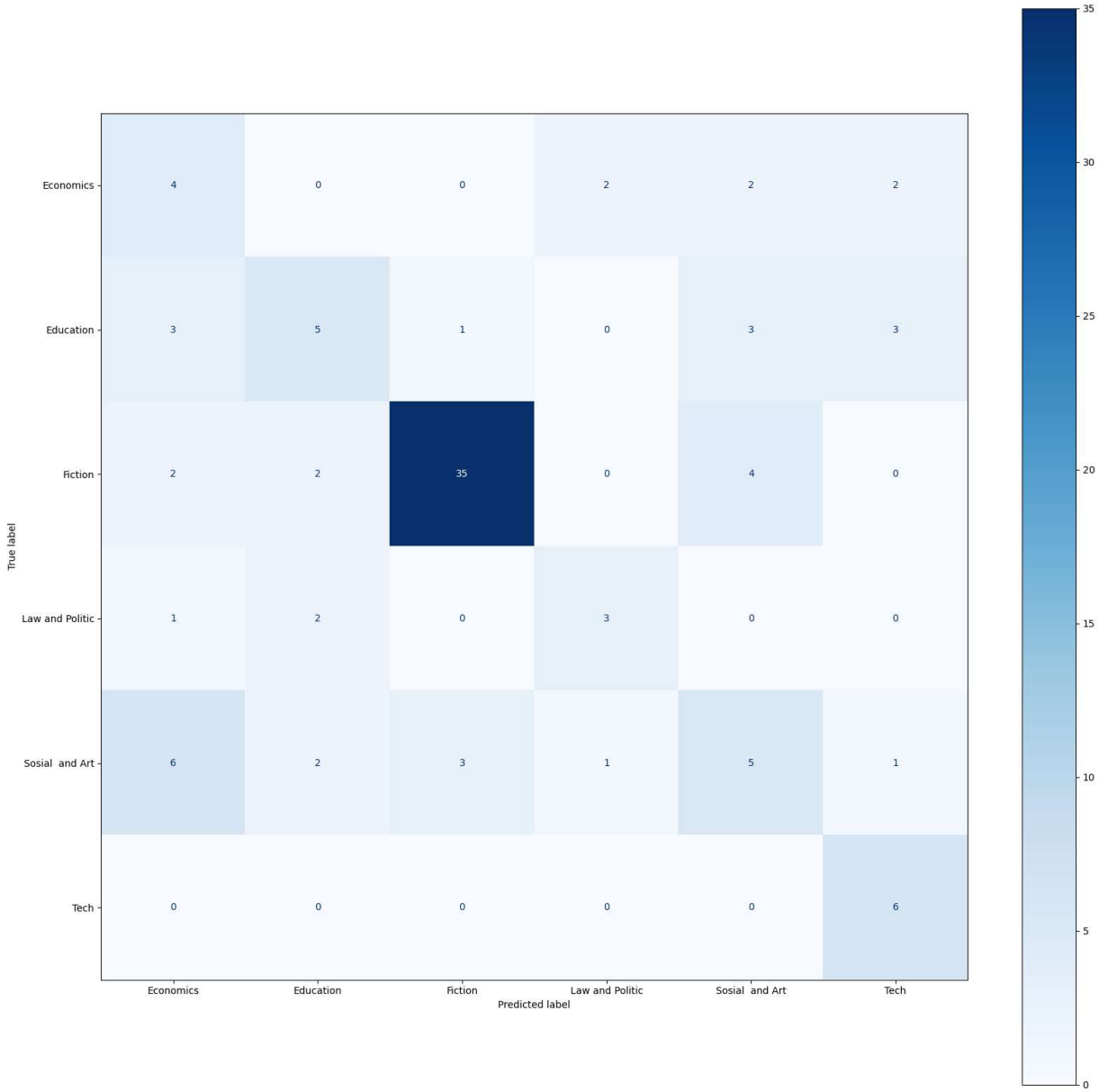


In [ ]:

```
print("NB Metrics:")

get_metrics(true_labels=test_labels, predicted_labels=predictions_NB)

print("Classification Report:")
print(metrics.classification_report(test_labels, predictions_NB))
print("Confusion Matrix:")
print(get_confusion_matrix(test_labels, predictions_NB))


disp = ConfusionMatrixDisplay(confusion_matrix=get_confusion_matrix(test_labels, pr

fig, ax = plt.subplots(figsize=(20, 20))

disp.plot(ax=ax, cmap=plt.cm.Blues)
```

```
NB Metrics:
Accuracy:  0.59
Precision:  0.62
Recall:  0.59
F1 Score:  0.59
Classification Report:
                 precision    recall  f1-score   support

      Economics       0.25      0.40      0.31        10
      Education       0.45      0.33      0.38        15
        Fiction       0.90      0.81      0.85        43
Law and Politic       0.50      0.50      0.50         6
Sosial  and Art       0.36      0.28      0.31        18
           Tech       0.50      1.00      0.67         6

       accuracy                           0.59        98
      macro avg       0.49      0.55      0.50        98
   weighted avg       0.62      0.59      0.59        98

Confusion Matrix:
[[ 4  0  0  2  2  2]
 [ 3  5  1  0  3  3]
 [ 2  2 35  0  4  0]
 [ 1  2  0  3  0  0]
 [ 6  2  3  1  5  1]
 [ 0  0  0  0  0  6]]
```

Out[ ]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x26853aa01d0>



Model Testing

Silakan isi bagian ini dengan script untuk keperluan testing model terbaik yang telah dihasilkan dengan memberikan input berupa data teks yang sama sekali baru dan belum ada di dataset.

In [ ]:

In [ ]: