

LimeTextClassification

November 30, 2025

Example of Sentiment Analysis using NusaX Dataset

1 Imports

```
[3]: import numpy as np
import pandas as pd
import ast
import matplotlib.pyplot as plt

import sklearn
import sklearn.datasets
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.pipeline import make_pipeline
from sklearn.metrics import f1_score, accuracy_score
```

```
[4]: import nltk
nltk.download('punkt')
nltk.download('punkt_tab')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
```

```
[4]: True
```

```
[5]: from nltk import word_tokenize
```

```
[6]: from lime.lime_text import LimeTextExplainer
```

```
[7]: from xaicsv import xai_csv_utils
```

2 Model Training

```
[10]: # grab the data first
!git clone https://github.com/IndoNLP/nusax.git
```

```
Cloning into 'nusax'...
remote: Enumerating objects: 301, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 301 (delta 4), reused 2 (delta 2), pack-reused 296 (from 1)
Receiving objects: 100% (301/301), 3.74 MiB | 9.41 MiB/s, done.
Resolving deltas: 100% (136/136), done.
```

```
[11]: def getxy_data(filedir):
      df = pd.read_csv(filedir, encoding='utf-8-sig')
      data = list(df['text'])
      return (data, list(df['label']))
```

```
[12]: directory = "nusax/datasets/sentiment/indonesian"
      X_train, y_train = getxy_data(directory + "/train.csv")
      X_valid, y_valid = getxy_data(directory + "/valid.csv")
      X_test, y_test = getxy_data(directory + "/test.csv")
```

```
[13]: X = X_train + X_valid
      y = y_train + y_valid
```

```
[14]: vectorizer = TfidfVectorizer(tokenizer=nlTK.word_tokenize)
      vectorizer = vectorizer.fit(X)
      X_v = vectorizer.transform(X)
      X_test_v = vectorizer.transform(X_test)
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/feature_extraction/text.py:517:
UserWarning: The parameter 'token_pattern' will not be used since 'tokenizer' is
not None'
      warnings.warn(
```

```
[15]: svc_linear = SVC(C=1, kernel='linear', probability=True)
      svc_linear = svc_linear.fit(X_v, y)
```

```
[16]: y_pred = svc_linear.predict(X_test_v)
      f1score = f1_score(y_test, y_pred, average='macro')
      print(f"F1 score : {f1score}")
```

```
F1 score : 0.7711602263856362
```

3 LIME

```
[17]: list_labels = ['negative', 'neutral', 'positive']
explainer = LimeTextExplainer(class_names=list_labels, random_state=42)
```

```
[18]: directory = "nusax/datasets/sentiment/indonesian/test.csv"
df_test = pd.read_csv(directory, encoding='utf-8-sig')
```

```
[19]: pipe = make_pipeline(vectorizer, svc_linear)
```

```
[21]: list_index_labels= list(range(0, len(list_labels)))

lime_weights = []
for i in range(len(X_test)):
    exp = explainer.explain_instance(X_test[i],
                                    pipe.predict_proba,
                                    num_features=6,
                                    labels= list_index_labels)

    weight_val = xai_csv_utils.lime_values_to_weights_dict(
                                                lime_explanation=exp,
                                                class_names=list_labels)

    lime_weights.append(weight_val)
```

```
[22]: df_test["predict_label"] = y_pred

predict_proba = svc_linear.predict_proba(X_test_v)
df_test["predict_proba"] = predict_proba.tolist()
df_test['lime_weights'] = lime_weights
```

```
[23]: df_test.head()
```

```
[23]:      id      text      label \
0  411  Dekat dengan hotel saya menginap, hanya ditemp...  positive
1  729      Iya benar, dia sedang jaga warung.  neutral
2  373  Kangkungnya lumayan tapi kepiting saus padangn...  negative
3  262  Bertempat di braga city walk yang satu gedung ...  positive
4  177  Gianyar terima bantuan sosial 2018 sebesar rp ...  neutral

      predict_label      predict_proba \
0      positive  [0.03714257204449854, 0.016908852303781936, 0...
1      neutral  [0.11812091128210017, 0.8595926895481094, 0.02...
2      negative  [0.6854055103734684, 0.19394585504816297, 0.12...
3      positive  [0.009206180531608288, 0.005159731677888622, 0...
4      neutral  [0.08506707870063696, 0.8698844858602772, 0.04...

      lime_weights
```

```

0 {'negative': [('hotel', 0.07531087709984967), ...
1 {'negative': [('dia', -0.20376838522115123), (...
2 {'negative': [('tidak', 0.30992904616248534), ...
3 {'negative': [('yang', -0.04105144252220528), ...
4 {'negative': [('terima', -0.12071193999232037)...

```

```
[24]: df_test.to_csv("nusax_results_lime.csv", sep=';', index=False)
```

4 Read LIME CSV

```
[25]: df = pd.read_csv("nusax_results_lime.csv", sep=";")
```

```
[26]: sample_weights = df.iloc[25]["lime_weights"]
      for label in (list_labels):
          fig = xai_csv_utils.weights_dict_to_pyplot(sample_weights, label)
```



