

LimeTabularClassification

November 30, 2025

1 Imports

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

import sklearn
import sklearn.datasets
import sklearn.ensemble
from sklearn.svm import SVC
from sklearn.pipeline import make_pipeline
from sklearn.metrics import f1_score, accuracy_score
```

```
[4]: import lime
import lime.lime_tabular
```

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

2 Model Training

```
[6]: iris = sklearn.datasets.load_iris()
train, test, labels_train, labels_test = sklearn.model_selection.
↳ train_test_split(iris.data, iris.target, random_state=42, train_size=0.80)
```

```
[7]: svc_iris = SVC(C=1, kernel='linear', probability=True)
```

```
[8]: svc_iris = svc_iris.fit(train, labels_train)
y_pred = svc_iris.predict(test)
f1score = f1_score(labels_test, y_pred, average='macro')
print (f"F1 score : {f1score}")
```

F1 score : 1.0

3 LIME

```
[9]: explainer = lime.lime_tabular.LimeTabularExplainer(train,
                                                         feature_names=iris.
                                                         ↪feature_names,
                                                         class_names=iris.
                                                         ↪target_names,
                                                         discretize_continuous=True)
```

```
[10]: df_test = pd.DataFrame(data=test, columns=iris.feature_names)
```

```
[12]: list_labels = iris.target_names.tolist()
list_index_labels= list(range(0, len(list_labels)))

lime_weights = []
for i in range(len(df_test)):
    exp = explainer.explain_instance(df_test.iloc[0].values,
                                     svc_iris.predict_proba,
                                     num_features=len(iris.feature_names),
                                     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)
```

```
[13]: df_test['label'] = labels_test
df_test["predict_label"] = y_pred

predict_probas = svc_iris.predict_proba(test)
df_test["predict_proba"] = predict_probas.tolist()
df_test['lime_weights'] = lime_weights
```

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

```
[14]:   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0                6.1                2.8                4.7                1.2
1                5.7                3.8                1.7                0.3
2                7.7                2.6                6.9                2.3
3                6.0                2.9                4.5                1.5
4                6.8                2.8                4.8                1.4

   label  predict_label  predict_proba  \
0      1              1  [0.006441199293076751, 0.9100250130483459, 0.0...
1      0              0  [0.9520177373223058, 0.0324201677517393, 0.015...
2      2              2  [2.8624475050043946e-06, 3.427996814735579e-06...
3      1              1  [0.007908265126189595, 0.9018027494572898, 0.0...
```

```
4      1      1 [0.005471769054130507, 0.8819369549659687, 0.1...
```

```
lime_weights
0 {'setosa': [('4.25 < petal length (cm) <= 5.10...
1 {'setosa': [('4.25 < petal length (cm) <= 5.10...
2 {'setosa': [('4.25 < petal length (cm) <= 5.10...
3 {'setosa': [('4.25 < petal length (cm) <= 5.10...
4 {'setosa': [('4.25 < petal length (cm) <= 5.10...
```

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

4 Read LIME CSV

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

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



