

ShapTabularClassification

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 shap
```

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

2 Model Training

```
[8]: 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)
```

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

```
[10]: 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 SHAP

```
[11]: explainer_shap = shap.KernelExplainer(svc_iris.predict_proba,
                                           train,
                                           feature_names = iris.feature_names)
shap_values = explainer_shap.shap_values(test)
```

WARNING:shap:Using 120 background data samples could cause slower run times.
Consider using shap.sample(data, K) or shap.kmeans(data, K) to summarize the
background as K samples.

```
0%|          | 0/30 [00:00<?, ?it/s]
```

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

```
[13]: list_labels = iris.target_names.tolist()

shap_weights = []
for i in range(len(shap_values)):
    weight_val = xai_csv_utils.
    ↪shap_values_to_weights_dict(class_names=list_labels,
                               classification=True,
                               shap_values=shap_values[i],
                               feature_names=explainer_shap.data_feature_names)

    shap_weights.append(weight_val)
```

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

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

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

```
[15]:   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  [0.006257046216995757, 0.9208064474879766, 0.0...
1      0              0  [0.9516051133665284, 0.03299234984171963, 0.01...
2      2              2  [2.8537894617380176e-06, 3.4175420641285076e-0...
3      1              1  [0.0077397558594794235, 0.9129178489803303, 0...
```

```
4      1      1 [0.005219234077771518, 0.8941870022330531, 0.1...
```

```
                                shap_weights
0 {'setosa': [('sepal length (cm)', -0.002328668...
1 {'setosa': [('sepal length (cm)', 0.0008243684...
2 {'setosa': [('sepal length (cm)', -0.005823765...
3 {'setosa': [('sepal length (cm)', -0.002051220...
4 {'setosa': [('sepal length (cm)', -0.006021350...
```

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

4 Read SHAP CSV

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

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



