

# ShapTabularRegression

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
from sklearn import svm

[4]: import shap

[5]: from xaicsv import xai_csv_utils
```

## 2 Model Training

```
[8]: diabetes = sklearn.datasets.load_diabetes()

train, test, labels_train, labels_test = sklearn.model_selection.
    ↪train_test_split(diabetes.data, diabetes.target, train_size=0.80)

[9]: regr = svm.SVR()

[10]: regr = regr.fit(train, labels_train)
y_pred = regr.predict(test)

[11]: print('SVR MSEError', np.mean((y_pred - labels_test) ** 2))
```

SVR MSEError 4392.465299434511

```
[12]: categorical_features = np.argwhere(np.array([len(set(diabetes.data[:,x])) for x in range(diabetes.data.shape[1])]) <= 10).flatten()
```

### 3 SHAP

```
[13]: # summarize train data for faster SHAP estimation values
train_summary = shap.kmeans(train, 50)
```

```
[14]: explainer_shap = shap.KernelExplainer(regr.predict,
                                           train_summary,
                                           feature_names = diabetes.feature_names)
shap_values = explainer_shap.shap_values(test)
```

```
0%|          | 0/89 [00:00<?, ?it/s]
```

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

```
[16]: list_labels = ["target"]

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

    shap_weights.append(weight_val)
```

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

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

```
[18]:
```

	age	sex	bmi	bp	s1	s2	s3	\
0	-0.078165	-0.044642	-0.016984	-0.012556	-0.000193	-0.013527	0.070730	
1	-0.103593	-0.044642	-0.037463	-0.026328	0.002559	0.019980	0.011824	
2	0.016281	0.050680	0.072474	0.076958	-0.008449	0.005575	-0.006584	
3	-0.027310	0.050680	-0.023451	-0.015999	0.013567	0.012778	0.026550	
4	-0.107226	-0.044642	-0.077342	-0.026328	-0.089630	-0.096198	0.026550	

  

	s4	s5	s6	label	predict_label	\
0	-0.039493	-0.041176	-0.092204	90.0	123.560373	
1	-0.002592	-0.068332	-0.025930	113.0	127.047950	
2	-0.002592	-0.023647	0.061054	131.0	154.170639	

```

3 -0.002592 -0.010903 -0.021788 71.0 122.514672
4 -0.076395 -0.042571 -0.005220 137.0 128.934332

```

```

                                shap_weights
0 {'target': [('age', 0.9776850211656871), ('sex...
1 {'target': [('age', 2.2235841651014354), ('sex...
2 {'target': [('age', 0.4059884867358234), ('sex...
3 {'target': [('age', -1.1546066222634046), ('se...
4 {'target': [('age', 1.6323981406616463), ('sex...

```

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

## 4 Read SHAP CSV

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

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

