

```

import warnings
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_curve, auc, classification_report
from sklearn.metrics import confusion_matrix
from sklearn.svm import SVC
warnings.filterwarnings("ignore", category=FutureWarning)
np.seterr(divide='ignore', invalid='ignore')
import eli5
from eli5.sklearn import PermutationImportance
import tensorflow.python.util.deprecation as deprecation
deprecation._PRINT_DEPRECATION_WARNINGS = False
import matplotlib.pyplot as plt
from pdpbox import pdp

```

```

warnings.filterwarnings("ignore")
dt = pd.read_csv("heart.csv")
desired_width = 520
pd.set_option('display.width', desired_width)

np.set_printoptions(linewidth=desired_width)

pd.set_option('display.max_columns', 25)

```

```

dt.columns = ['age', 'sex', 'chest_pain_type', 'resting_blood_pressure', 'cholesterol', 'fasting_blood_sugar', 'rest_ecg', 'exercise_induced_angina']
dt['sex'][dt['sex'] == 0] = 'female'
dt['sex'][dt['sex'] == 1] = 'male'
#print(len(dt.columns))
dt['chest_pain_type'][dt['chest_pain_type'] == 0] = 'typical angina'
dt['chest_pain_type'][dt['chest_pain_type'] == 1] = 'atypical angina'
dt['chest_pain_type'][dt['chest_pain_type'] == 2] = 'non-anginal pain'
dt['chest_pain_type'][dt['chest_pain_type'] == 3] = 'asymptomatic'

dt['fasting_blood_sugar'][dt['fasting_blood_sugar'] == 0] = 'lower than 120mg/ml'
dt['fasting_blood_sugar'][dt['fasting_blood_sugar'] == 1] = 'greater than 120mg/ml'

dt['rest_ecg'][dt['rest_ecg'] == 0] = 'normal'
dt['rest_ecg'][dt['rest_ecg'] == 1] = 'ST-T wave abnormality'
dt['rest_ecg'][dt['rest_ecg'] == 2] = 'left ventricular hypertrophy'

dt['exercise_induced_angina'][dt['exercise_induced_angina'] == 0] = 'no'
dt['exercise_induced_angina'][dt['exercise_induced_angina'] == 1] = 'yes'

```

```

dt['st_slope'][dt['st_slope'] == 0] = 'upsloping'
dt['st_slope'][dt['st_slope'] == 1] = 'flat'
dt['st_slope'][dt['st_slope'] == 2] = 'downsloping'

dt['thalassemia'][dt['thalassemia'] == 1] = 'normal'
dt['thalassemia'][dt['thalassemia'] == 2] = 'fixed defect'
dt['thalassemia'][dt['thalassemia'] == 3] = 'reversable defect'

dt['sex'] = dt['sex'].astype('object')
dt['chest_pain_type'] = dt['chest_pain_type'].astype('object')
dt['fasting_blood_sugar'] = dt['fasting_blood_sugar'].astype('object')
dt['rest_ecg'] = dt['rest_ecg'].astype('object')
dt['exercise_induced_angina'] = dt['exercise_induced_angina'].astype('object')
dt['st_slope'] = dt['st_slope'].astype('object')
dt['thalassemia'] = dt['thalassemia'].astype('object')

```

```

dt = pd.get_dummies(dt,prefix=['st_slope'],columns=['st_slope'])
dt = pd.get_dummies(dt, drop_first=True)

```

```

X_train, X_test, y_train, y_test = train_test_split(dt.drop('target', 1), dt['target'])
model = SVC(kernel='linear',gamma='scale',probability=True)
model.fit(X_train, y_train)

```

```

☞ SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
      decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear',
      max_iter=-1, probability=True, random_state=None, shrinking=True, tol=0.001,
      verbose=False)

```

```

acc = model.score(X_test,y_test)*100
print("Accuracy = ",acc)

```

```

y_predict = model.predict(X_test)
y_pred_quant = model.predict_proba(X_test)[: , 1]
y_pred_bin = model.predict(X_test)

```

```

confusion_matrix = confusion_matrix(y_test, y_pred_bin)
total=sum(sum(confusion_matrix))

```

```

sensitivity = confusion_matrix[0,0]/(confusion_matrix[0,0]+confusion_matrix[1,0])
print('Sensitivity : ', sensitivity )

```

```

specificity = confusion_matrix[1,1]/(confusion_matrix[1,1]+confusion_matrix[0,1])

```

```
print('Specificity : ', specificity)
```

```
↳ Accuracy = 87.95180722891565  
Sensitivity : 0.8717948717948718  
Specificity : 0.8863636363636364
```

```
print("Higher the AUC, better the model is at predicting,", "Score above 0.90 is cons  
fpr, tpr, thresholds = roc_curve(y_test, y_pred_quant)  
print(auc(fpr, tpr))
```

```
↳ Higher the AUC, better the model is at predicting, Score above 0.90 is consider  
0.9172494172494172
```

```
base_features = dt.columns.values.tolist()  
base_features.remove('target')
```

```
feat_name = 'num_major_vessels'  
pdp_dist = pdp.pdp_isolate(model=model, dataset=X_test, model_features=base_features
```

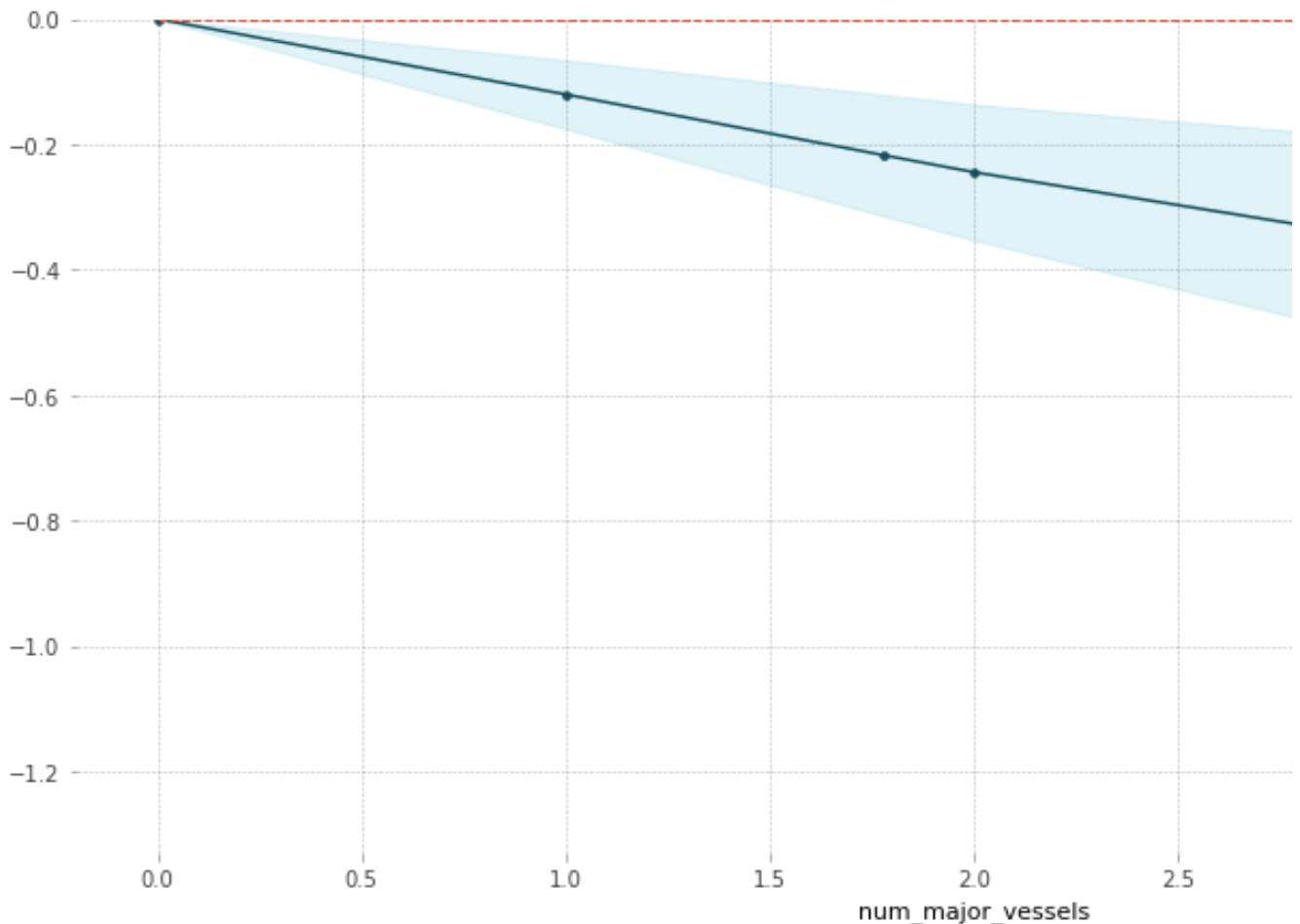
```
pdp.pdp_plot(pdp_dist, feat_name)  
plt.show()
```

```
↳
```

```
findfont: Font family ['Arial'] not found. Falling back to DejaVu Sans.  
findfont: Font family ['Arial'] not found. Falling back to DejaVu Sans.  
findfont: Font family ['Arial'] not found. Falling back to DejaVu Sans.  
findfont: Font family ['Arial'] not found. Falling back to DejaVu Sans.
```

PDP for feature "num_major_vessels"

Number of unique grid points: 5



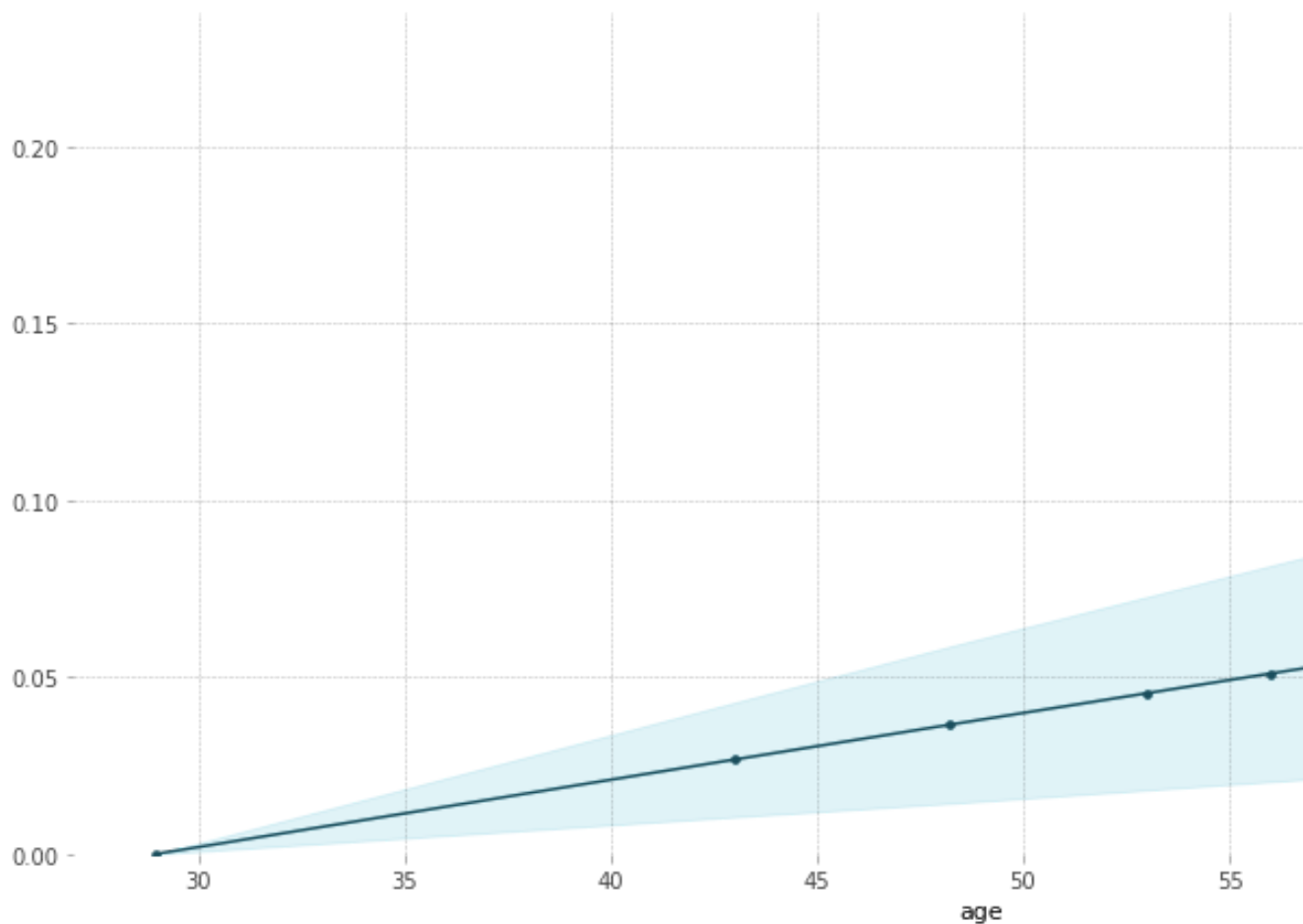
```
feat_name = 'age'  
pdp_dist = pdp.pdp_isolate(model=model, dataset=X_test, model_features=base_features
```

```
pdp.pdp_plot(pdp_dist, feat_name)  
plt.show()
```



PDP for feature "age"

Number of unique grid points: 10



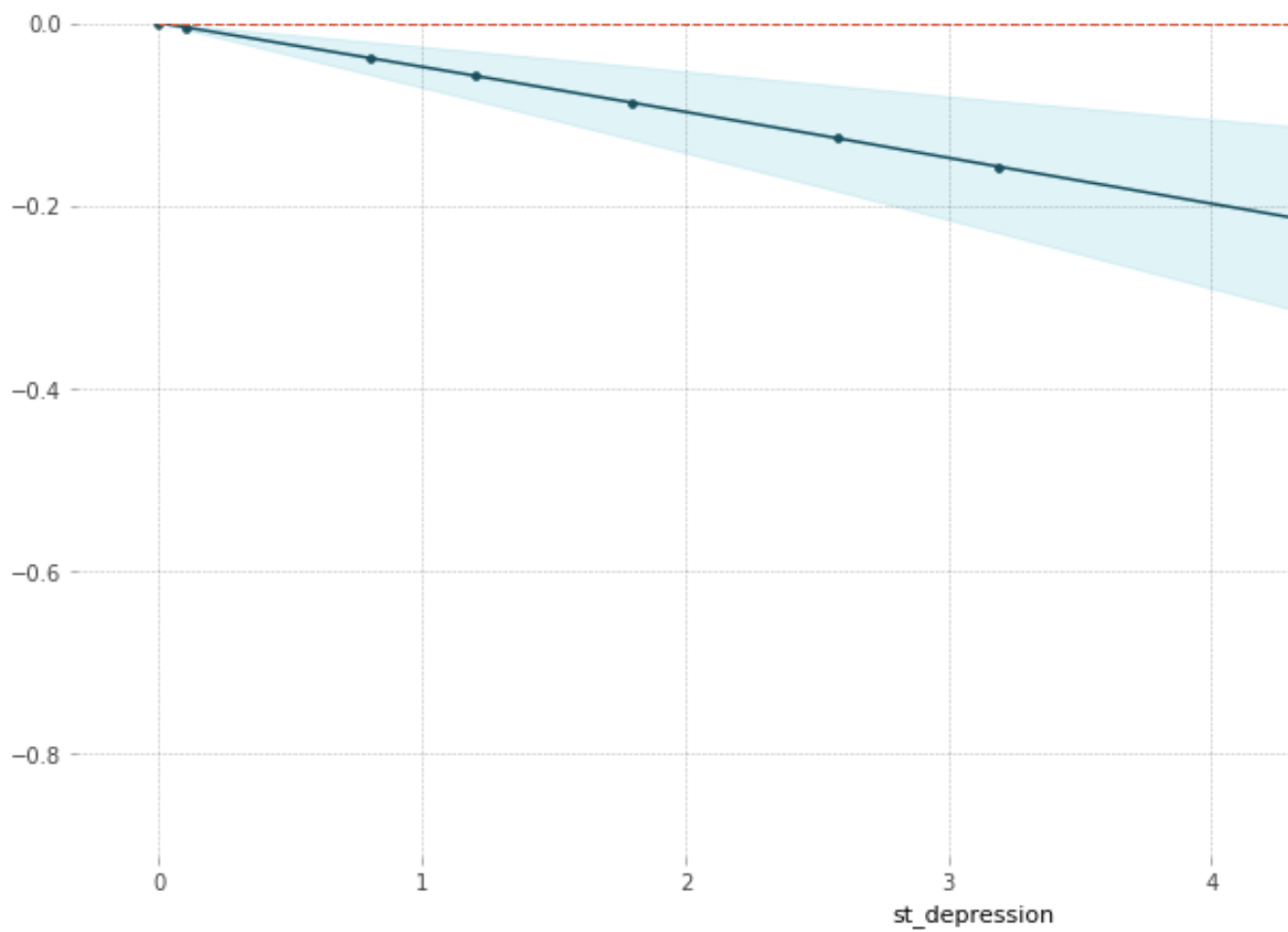
```
feat_name = 'st_depression'  
pdp_dist = pdp.pdp_isolate(model=model, dataset=X_test, model_features=base_features
```

```
pdp.pdp_plot(pdp_dist, feat_name)  
plt.show()
```



PDP for feature "st_depression"

Number of unique grid points: 8



```
perm = PermutationImportance(model, random_state=1).fit(X_test, y_test)
diag = eli5.explain_weights(perm, feature_names = X_test.columns.tolist())
#print(eli5.formatters.format_as_dataframe(diag))
diag
```



Weight	Feature
0.0482 ± 0.0806	num_major_vessels
0.0386 ± 0.0415	chest_pain_type_typical angina
0.0265 ± 0.0180	thalassemia_fixed defect
0.0241 ± 0.0152	resting_blood_pressure
0.0193 ± 0.0447	st_depression
0.0169 ± 0.0246	thalassemia_reversable defect
0.0169 ± 0.0246	max_heart_rate_achieved
0.0145 ± 0.0180	st_slope_flat
0.0145 ± 0.0180	chest_pain_type_atypical angina
0.0120 ± 0.0000	thalassemia_normal
0.0072 ± 0.0193	age
0.0048 ± 0.0118	st_slope_upsloping
0.0048 ± 0.0118	exercise_induced_angina_yes
0.0048 ± 0.0193	st_slope_downsloping
0 ± 0.0000	chest_pain_type_non-anginal pain
0 ± 0.0000	fasting_blood_sugar_lower than 120mg/ml
0 ± 0.0000	rest_ecg_left ventricular hypertrophy
0 ± 0.0000	rest_ecg_normal
0 ± 0.0000	cholesterol
-0.0000 ± 0.0264	sex_male

```
cr = classification_report(y_test,y_predict)
print(cr)
```

	precision	recall	f1-score	support
0	0.87	0.87	0.87	39
1	0.89	0.89	0.89	44
accuracy			0.88	83
macro avg	0.88	0.88	0.88	83
weighted avg	0.88	0.88	0.88	83

