

```
pip install lime
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting lime
  Downloading lime-0.2.0.1.tar.gz (275 kB)
    |████████████████████████████████████████| 275 kB 5.9 MB/s
Requirement already satisfied: matplotlib in /usr/local/lib/python3.8/dist-packages (from lime) (3.2.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.8/dist-packages (from lime) (1.21.6)
Requirement already satisfied: scipy in /usr/local/lib/python3.8/dist-packages (from lime) (1.7.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.8/dist-packages (from lime) (4.64.1)
Requirement already satisfied: scikit-learn>=0.18 in /usr/local/lib/python3.8/dist-packages (from lime) (1.0.2)
Requirement already satisfied: scikit-image>=0.12 in /usr/local/lib/python3.8/dist-packages (from lime) (0.18.3)
Requirement already satisfied: networkx>=2.0 in /usr/local/lib/python3.8/dist-packages (from scikit-image>=0.12->lime) (2.8.8)
Requirement already satisfied: pillow!=7.1.0,!>=7.1.1,>=4.3.0 in /usr/local/lib/python3.8/dist-packages (from scikit-image>=0.12->lime) (9.5.0)
Requirement already satisfied: PyWavelets>=1.1.1 in /usr/local/lib/python3.8/dist-packages (from scikit-image>=0.12->lime) (1.4.1)
Requirement already satisfied: imageio>=2.3.0 in /usr/local/lib/python3.8/dist-packages (from scikit-image>=0.12->lime) (2.9.0)
Requirement already satisfied: tifffile>=2019.7.26 in /usr/local/lib/python3.8/dist-packages (from scikit-image>=0.12->lime) (2022.9.10)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib->lime) (1.4.4)
Requirement already satisfied: pyparsing!=2.0.4,!>=2.1.2,!>=2.1.6,>=2.0.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib->lime) (3.0.9)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.8/dist-packages (from matplotlib->lime) (0.11.0)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib->lime) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-packages (from python-dateutil>=2.1->matplotlib->lime) (1.16.0)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.8/dist-packages (from scikit-learn>=0.18->lime) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.8/dist-packages (from scikit-learn>=0.18->lime) (3.1.0)
Building wheels for collected packages: lime
  Building wheel for lime (setup.py) ... done
  Created wheel for lime: filename=lime-0.2.0.1-py3-none-any.whl size=283857 sha256=b05a8a640095d45dcf56d7ed81107b3a9338b856feeaf5f
  Stored in directory: /root/.cache/pip/wheels/e6/a6/20/cc1e293fcd67ede666fed293cb895395e7ecceb4467779546
Successfully built lime
Installing collected packages: lime
Successfully installed lime-0.2.0.1
```

```
pip install shap
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting shap
  Downloading shap-0.41.0-cp38-cp38-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (575 kB)
    |████████████████████████████████████████| 575 kB 7.8 MB/s
Requirement already satisfied: cloudpickle in /usr/local/lib/python3.8/dist-packages (from shap) (1.5.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.8/dist-packages (from shap) (1.21.6)
Requirement already satisfied: tqdm>4.25.0 in /usr/local/lib/python3.8/dist-packages (from shap) (4.64.1)
Requirement already satisfied: numba in /usr/local/lib/python3.8/dist-packages (from shap) (0.56.4)
Collecting slicer==0.0.7
  Downloading slicer-0.0.7-py3-none-any.whl (14 kB)
Requirement already satisfied: packaging>20.9 in /usr/local/lib/python3.8/dist-packages (from shap) (21.3)
Requirement already satisfied: pandas in /usr/local/lib/python3.8/dist-packages (from shap) (1.3.5)
Requirement already satisfied: scipy in /usr/local/lib/python3.8/dist-packages (from shap) (1.7.3)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.8/dist-packages (from shap) (1.0.2)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.8/dist-packages (from packaging>20.9->shap) (3.0.9)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.8/dist-packages (from numba->shap) (4.13.0)
Requirement already satisfied: llvmlite<0.40,>=0.39.0dev0 in /usr/local/lib/python3.8/dist-packages (from numba->shap) (0.39.1)
Requirement already satisfied: setuptools in /usr/local/lib/python3.8/dist-packages (from numba->shap) (57.4.0)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.8/dist-packages (from importlib-metadata->numba->shap) (3.11.0)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.8/dist-packages (from pandas->shap) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.8/dist-packages (from pandas->shap) (2022.6)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-packages (from python-dateutil>=2.7.3->pandas->shap) (1.16.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.8/dist-packages (from scikit-learn->shap) (3.1.0)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.8/dist-packages (from scikit-learn->shap) (1.2.0)
Installing collected packages: slicer, shap
Successfully installed shap-0.41.0 slicer-0.0.7
```

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_breast_cancer
import xgboost as xgb
from sklearn.metrics import accuracy_score
import shap

# Read the DataFrame, first using the feature data

data = load_breast_cancer()

df = pd.DataFrame(data.data, columns=data.feature_names)

# Add a target column, and fill it with the target data

df['target'] = data.target

# Set up the data for modelling

y=df['target'].to_frame() # define Y
```

```

X=df[df.columns.difference(['target'])] # define X

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=42) # create train and test

# build model - Xgboost

xgb_mod=xgb.XGBClassifier(random_state=42,gpu_id=0) # build classifier Gradient Boosted decision trees

xgb_mod=xgb_mod.fit(X_train,y_train.values.ravel())

# make prediction and check model accuracy

y_pred = xgb_mod.predict(X_test)

# Performance

accuracy = accuracy_score(y_test, y_pred)

print("Accuracy: %.2f%%" % (accuracy * 100.0))

    Accuracy: 95.61%

#LIME

# Utilizing our same xgb_mod model object created above

# Import pacakages

import lime

import lime.lime_tabular

import numpy as np

import xgboost

##### create explainer #####

# we use the dataframes splits created above for SHAP

explainer = lime.lime_tabular.LimeTabularExplainer(X_test.to_numpy(), feature_names=X_test.columns, class_names=['0','1'], verbose=True)

##### visualizations #####

exp = explainer.explain_instance(X_test.iloc[0], xgb_mod.predict_proba)#, num_features=20)

exp.show_in_notebook(show_table=True)

-----
ValueError                                Traceback (most recent call last)
<ipython-input-4-c2c999f4bb0e> in <module>
    23 ##### visualizations #####
    24
--> 25 exp = explainer.explain_instance(X_test.iloc[0], xgb_mod.predict_proba)#,
num_features=20)
    26
    27 exp.show_in_notebook(show_table=True)

3 frames
/usr/local/lib/python3.8/dist-packages/xgboost/core.py in _validate_features(self,
data)
    1687         ', '.join(str(s) for s in my_missing))
    1688
-> 1689         raise ValueError(msg.format(self.feature_names,
    1690                                   data.feature_names))
    1691

ValueError: feature_names mismatch: ['area error', 'compactness error', 'concave
points error', 'concavity error', 'fractal dimension error', 'mean area', 'mean
compactness', 'mean concave points', 'mean concavity', 'mean fractal dimension',
'mean perimeter', 'mean radius', 'mean smoothness', 'mean symmetry', 'mean texture',
'perimeter error', 'radius error', 'smoothness error', 'symmetry error', 'texture
error', 'worst area', 'worst compactness', 'worst concave points', 'worst
concavity', 'worst fractal dimension', 'worst perimeter', 'worst radius', 'worst

```

#SHAP

```
# Generate the Tree explainer and SHAP values

explainer = shap.TreeExplainer(xgb_mod) #fast and exact method to estimate SHAP values for tree models and ensembles of trees,

shap_values = explainer.shap_values(X)

expected_value = explainer.expected_value

##### visualizations #####

# Generate summary dot plot

shap.summary_plot(shap_values, X,title="SHAP summary plot")

# Generate summary bar plot

shap.summary_plot(shap_values, X,plot_type="bar")

# Generate waterfall plot

shap.plots._waterfall.waterfall_legacy(expected_value, shap_values[79], features=X.loc[79,:], feature_names=X.columns, max_display=15, st

# Generate dependence plot

shap.dependence_plot("worst concave points", shap_values, X, interaction_index="mean concave points")

# Generate multiple dependence plots

for name in X_train.columns:

    shap.dependence_plot(name, shap_values, X)

shap.dependence_plot("worst concave points", shap_values, X, interaction_index="mean concave points")

# Generate force plot - Multiple rows

shap.force_plot(explainer.expected_value, shap_values[:100,:], X.iloc[:100,:])

# Generate force plot - Single

shap.force_plot(explainer.expected_value, shap_values[0,:], X.iloc[0,:])

# Generate Decision plot

shap.decision_plot(expected_value, shap_values[79],link='logit' ,features=X.loc[79,:], feature_names=(X.columns.tolist()),show=True,title=
```