Python Transformerを利用した
カーネル密度分布の作成

KSK ANALYTICS

1. カーネル密度推定

カーネル密度推定による可視化

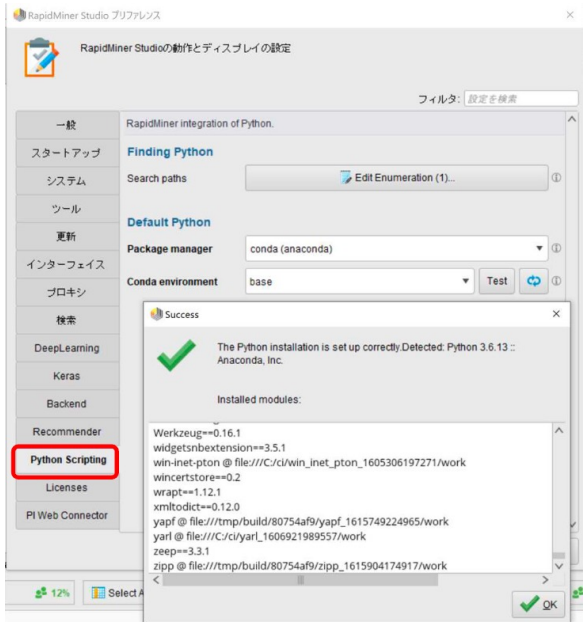
散布図は、2つの要素間の関係性を視覚的に把握する際に役立ちますが、プロットされる点の数が数百、数千を超える場合、点が重なり合いデータ間の関係性が把握しづらいという問題点があります。

カーネル密度推定（KDE）は、1次元または複数次元の連続した確率密度曲線を用いてデータを表現します。カーネル密度推定の手法を使うことで、データの分布を密度で把握することができるため、データ全体の傾向をより取れやすくなります。

RapidMinerには、カーネル密度推定を用いた可視化機能を備えていないため、本稿では、Python Extensionを用いて、カーネル密度推定による可視化機能を実装します。

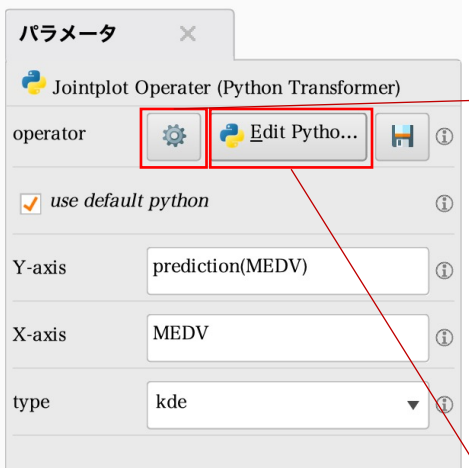
2．Pythonの設定

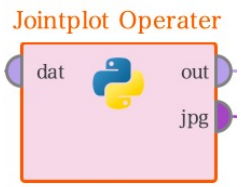
インストール後、設定 ＞ プリファレンス よりPythonの設定を行います。
Python Scriptingタブをクリックし、使用する環境を選択します。
Test をクリックして、OKならその環境を使用できます。
エラーが出た場合は、ライブラリのインストールやアップデートを試してください。



※RapidMiner Studio 9.9以上、Python Scripting Extension 9.9.1以上

3. 実装例





Python Script

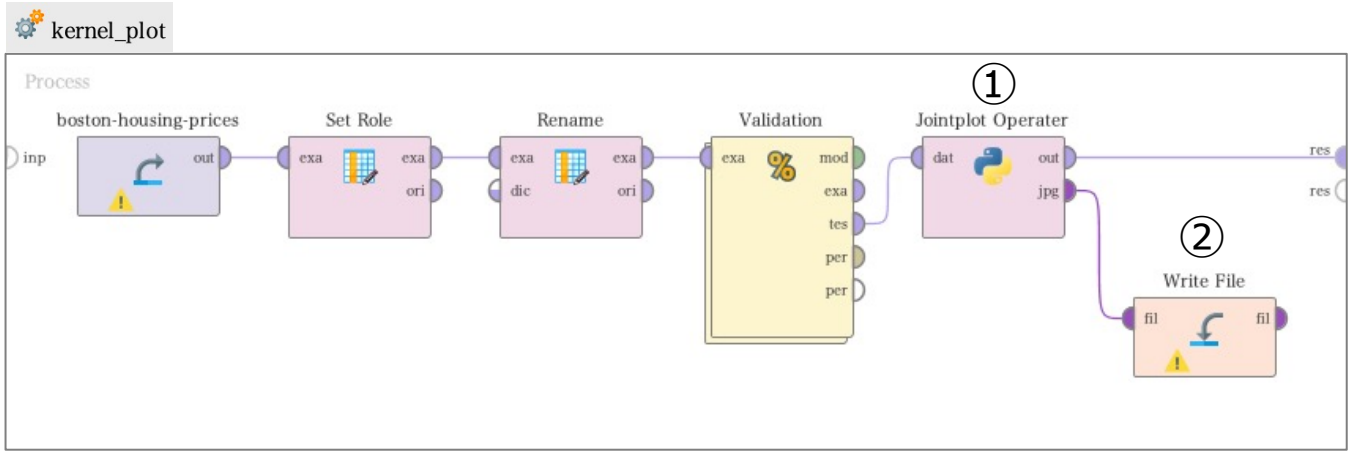
This Python script defines the operator logic. The method **rm_main(..)** will be used as entry point.

```python
import matplotlib.pyplot as plt
import seaborn as sns

def rm_main(data, parameters):
    g = sns.jointplot(x=parameters["X-axis"], y=parameters["Y-axis"], data=data, kind=parameters["type"])
    g.plot_joint(plt.scatter, c="k", s=10, linewidth=1, marker="+")
    g.ax_joint.collections[0].set_alpha(0)
    g.set_axis_labels(parameters["X-axis"], parameters["Y-axis"])
    path = "test.jpg"
    g.savefig(path)
    myfile = open(path,'r')
    # return data set and file
    return(data, myfile)
```

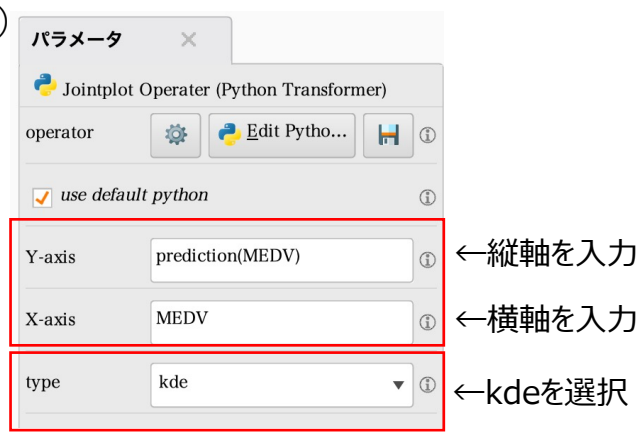
```json
{
  "name": "Jointplot Operater",
  "dropSpecial": false,
  "parameters": [
    {
      "name": "Y-axis",
      "type": "string",
      "description":  "Enter the Y axis name.",
      "optional": true
    },
    {
      "name": "X-axis",
      "type": "string",
      "description": "Enter the X axis name.",
      "optional": true
    },
    {
      "name": "type",
      "type": "category",
      "description": "An example of a categorical parameter.",
      "categories": [
        "scatter",
        "reg",
        "resid",
        "kde",
        "hex"
      ],
      "value": "kde"
    }
  ],
  "inputs": [
    {
      "name": "data",
      "type": "table"
    }
  ],
  "outputs": [
    {
      "name": "out",
      "type": "table"
    },
    {
      "name": "jpg",
      "type": "file"
    }
  ]
}
```

4. プロセス



ボストンの住宅価格を予測するモデルを作成し、モデルによる予測値と実測値の関係性をカーネル密度推定を使って可視化してみます。

①



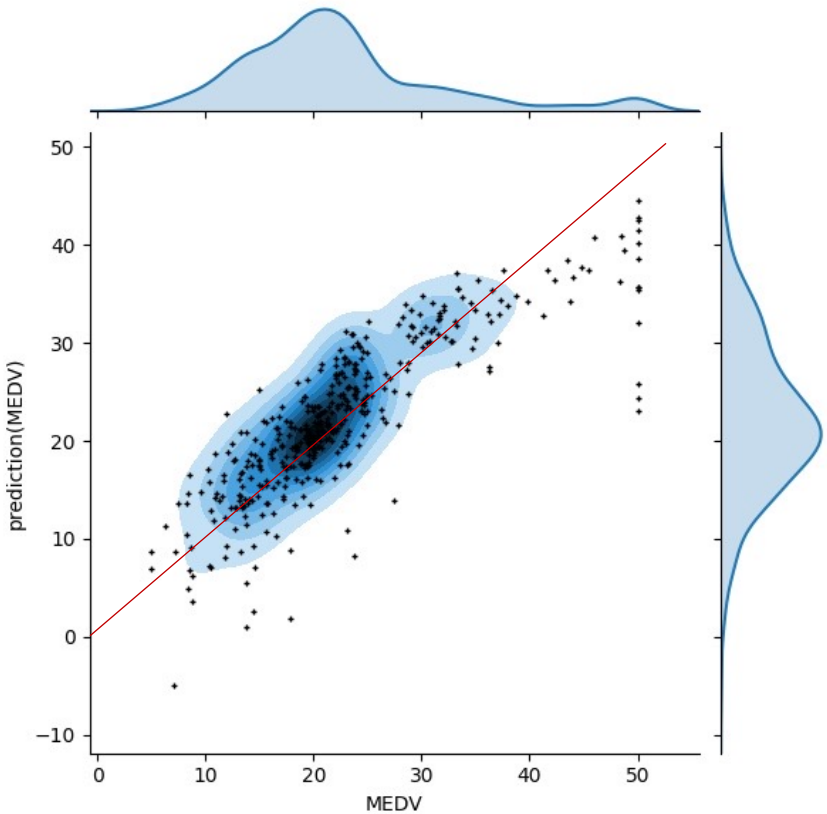
②

5．Prediction Chart

予測結果

Row No.	MEDV	prediction(MEDV)	CRIM: per c...	ZN: proport...	INDUS: pro...	CHAS:
1	27.500	25.085	0.149	0	8.560	0
2	30.800	31.191	0.028	75	2.950	0
3	13.100	14.513	1.152	0	8.140	0
4	20	23.126	0.102	0	12.830	0
5	30.100	34.868	0.657	20	3.970	0
6	24.800	26.435	0.043	52.500	5.320	0
7	14.500	18.506	8.492	0	18.100	0
8	19.500	17.877	0.171	0	8.560	0
9	13.400	13.486	7.050	0	18.100	0
10	13.300	16.701	6.393	0	18.100	0
11	5	8.722	67.921	0	18.100	0
12	23.100	10.814	13.522	0	18.100	0
13	23.600	30.958	0.057	0	3.410	0
14	13.800	11.476	2.379	0	19.580	0
15	20.700	27.841	0.091	20	6.960	1
16	31.600	32.519	0.412	0	6.200	0

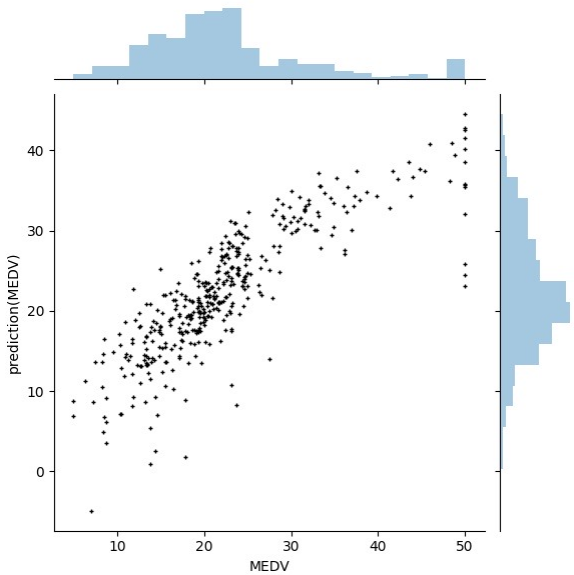
実行結果



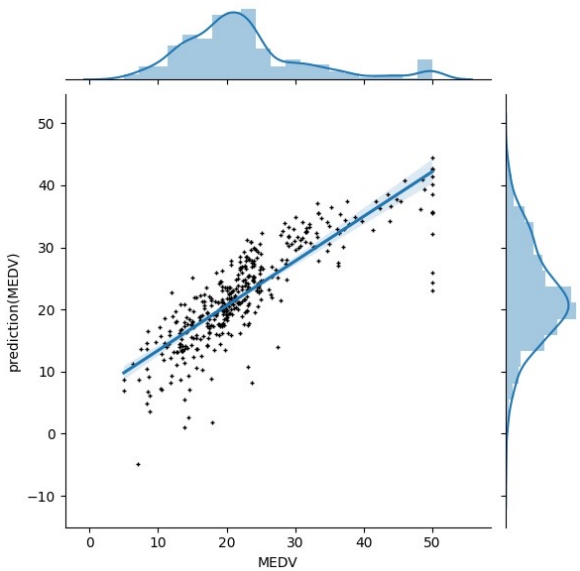
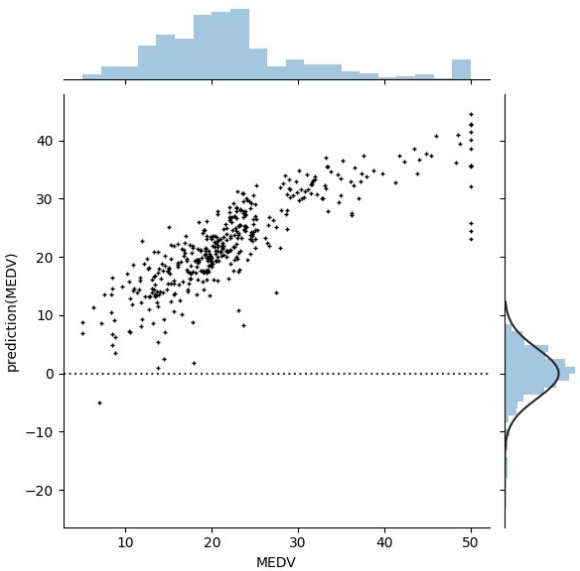
RMSE: 4.862 +/- 1.139
R^2 : 0.732 +/- 0.096

参考

Scatter



Reg



Resid