

## Feature-engine

```
pip install feature-engine
```

```
Collecting feature-engine
  Downloading feature_engine-1.6.2-py2.py3-none-any.whl (328 kB)
    328.9/328.9 kB 2.1 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.18.2 in /usr/local/lib/python3.10/dist-packages (from feature-engine) (1.23.5)
Requirement already satisfied: pandas>=1.0.3 in /usr/local/lib/python3.10/dist-packages (from feature-engine) (1.5.3)
Requirement already satisfied: scikit-learn>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from feature-engine) (1.2.2)
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.10/dist-packages (from feature-engine) (1.11.4)
Requirement already satisfied: statsmodels>=0.11.1 in /usr/local/lib/python3.10/dist-packages (from feature-engine) (0.14.1)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0.3->feature-engine) (2.8.1)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0.3->feature-engine) (2023.3)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.0.0->feature-engine) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.0.0->feature-engine) (3.2.0)
Requirement already satisfied: patsy>=0.5.4 in /usr/local/lib/python3.10/dist-packages (from statsmodels>=0.11.1->feature-engine) (0.5.4)
Requirement already satisfied: packaging>=21.3 in /usr/local/lib/python3.10/dist-packages (from statsmodels>=0.11.1->feature-engine) (23.1)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from patsy>=0.5.4->statsmodels>=0.11.1->feature-engine) (1.16.0)
Installing collected packages: feature-engine
Successfully installed feature-engine-1.6.2
```

```
import pandas as pd
import numpy as np
from feature_engine.imputation import MeanMedianImputer
from feature_engine.imputation import ArbitraryNumberImputer
```

```
X = pd.DataFrame(dict(
    x1 = [np.nan, 1, 1, 0, np.nan],
    x2 = ["a", np.nan, "b", np.nan, "a"],
))
mmi = MeanMedianImputer(imputation_method='median')
mmi.fit(X)
mmi.transform(X)
```

	x1	x2
0	1.0	a
1	1.0	NaN
2	1.0	b
3	0.0	NaN
4	1.0	a

```
#linear regression Feature importance
from sklearn.datasets import make_regression
from sklearn.linear_model import LinearRegression
from matplotlib import pyplot
```

```
#define dataset
X,y = make_regression(n_samples = 1000, n_features = 10, n_informative = 5, random_state=1)
print(X)
print(y)
```



```
-1.85522340e+02 -1.46659525e+02 1.90527552e+02 -1.78451438e+02
5.74887734e+01 6.22650526e+01 -2.92632921e+02 8.97522801e+01
-1.91354522e+01 -1.64150085e+02 -1.87321662e+02 -7.44857575e+01
-8.04354162e+01 6.86452731e+01 -1.55766601e+01 9.18264690e+01
6.67782077e+01 9.07662401e+01 1.05390570e+02 -1.75013182e+02
-6.60974494e+01 -9.06738270e+01 1.57725742e+02 -1.65912914e+02
-5.64399414e+01 -6.65081460e+01 -7.26406181e+01 -5.10926612e+01
-3.49734549e+01 3.55960072e+01 1.61116562e+02 5.67937636e+01
-2.35856803e+02 1.22714706e+01 -7.60215593e+01 3.29527602e+02
-1.12163956e+02 4.74364226e+01 1.41043812e+02 1.20192624e+02
3.17254522e+01 -8.06723696e+01 1.67454591e+02 -4.48007186e+01
8.06408356e+00 1.41211761e+02 -1.41676026e+02 -1.13483950e+01
2.47618814e+01 9.12720109e+01 8.60001871e+01 1.75376031e+02
-5.83973677e+00 2.03129861e+02 1.55768727e+02 1.50890858e+02
7.56363249e+00 -1.22252092e+01 -9.23299302e+01 6.44178620e+01
1.27934359e+02 3.22574232e+02 9.25429377e+01 -8.50578660e+01
-4.21236353e+01 8.07969166e+01 1.76883323e+02 5.96441341e+01
-1.86196788e+02 1.98198986e+02 9.29990604e+01 -9.66954765e+01
-1.12997672e+02 -1.47062667e+02 -1.44604461e+01 -5.55209941e+01
-1.53948970e+02 -8.20590622e+01 -5.20716713e+01 -1.10074497e+02
2.24021702e+02 -1.66007119e+01 2.24193193e+02 1.92250976e+01
2.73847938e+02 -7.32042966e+01 -8.52798932e+01 -9.48820083e+01
2.21717791e+01 -1.61220482e+02 1.45967873e+02 -3.09994276e+02
-2.02028034e+02 -1.12219977e+02 -2.08661308e+02 1.08242338e+02
-1.47533564e+02 -1.29558556e+02 1.71483047e+01 -5.99546433e+01
-6.60454409e+01 -8.08894870e+01 -2.17055691e+02 1.44635884e+02
1.08663240e+02 -3.62222925e+00 -1.21494745e+02 -5.61510065e+01
-1.94774346e+02 1.05574500e+02 1.69434490e+02 2.53386403e+01
-2.49879691e+02 5.47612021e+01 -3.89924018e+00 5.98817745e+01
-5.35324973e+01 1.36424075e+01 4.36980268e+01 8.68697845e+01
1.41802193e+02 1.56101208e+02 8.02513759e+01 -1.14317614e+01
1.23067318e+02 -4.83287115e+01 -2.21956103e+01 -7.48781802e+01
2.30816006e+02 4.51240930e+01 -6.15724583e+01 -3.50107772e+02
7.33796770e+01 7.97852933e+01 1.32517589e+01 -6.21098841e-02
-1.00649480e+02 1.44254869e+02 2.35376670e+01 1.17149745e+01
-1.10885117e+02 2.17778008e+02 5.25752157e+01 1.12813101e+02
```

```
# define the model
model = LinearRegression()
```

```
#fit the model
model.fit(X,y)
```

```
▼ LinearRegression
LinearRegression()
```

```
#get importance
importance = model.coef_
```

```
#summarize feature importance
for i,v in enumerate(importance):
    print('Feature: %0d, Score: %.5f' % (i,v))
```

```
Feature: 0, Score: 0.00000
Feature: 1, Score: 12.44483
Feature: 2, Score: -0.00000
Feature: 3, Score: -0.00000
Feature: 4, Score: 93.32225
Feature: 5, Score: 86.50811
Feature: 6, Score: 26.74607
Feature: 7, Score: 3.28535
Feature: 8, Score: -0.00000
Feature: 9, Score: 0.00000
```

```
#plot feature importance
pyplot.bar([x for x in range(len(importance))], importance)
```

