example_featureimportance

February 3, 2022

Toolset for calculating feature importance based on multiple methods:

- Hierarchical clustered Spearman correlation diagram
- linear/log-scaled Bayesian Linear Regression
- Random Forest Permutation Importance
- Model-agnostic correlation coefficients (see "A new coefficient of correlation" Chatterjee, S. (2019, September 22)

This script can also generate synthetic data and includes tests for all methods, which can be used to compare the results.

User settings, such as input/output paths and all other options, are set in the settings file (Default filename: settings_featureimportance.yaml) Alternatively, the settings file can be specified as a command line argument with: '-s', or '–settings' followed by PATH-TO-FILE/FILENAME.yaml (e.g. python featureimportance.py -s settings_featureimportance.yaml).

Requirements: - python>=3.9 - matplotlib>=3.5.1 - numpy>=1.22.0 - pandas>=1.3.5 - PyYAML>=6.0 - scikit_learn>=1.0.2 - scipy>=1.7.3 - xicor>=1.0.1

For more package details see conda environment file: environment.yaml

This package is part of the machine learning project developed for the Agricultural Research Federation (AgReFed).

[5]:
```
import os
import itertools
import sys
import yaml
import shutil
import argparse
from types import SimpleNamespace
import numpy as np
import pandas as pd
from sklearn.datasets import make_regression
from sklearn.preprocessing import StandardScaler, MinMaxScaler,␣
 ↪PowerTransformer, RobustScaler
from sklearn.linear_model import BayesianRidge
from sklearn.ensemble import RandomForestRegressor
```

```
from sklearn.inspection import permutation_importance
from scipy.stats import spearmanr
from scipy.cluster import hierarchy
import matplotlib as mpl
import matplotlib.pyplot as plt
#from xicor.xicor import Xi

!pip install git+git://github.com/czbiohub/xicor/
```

```
Collecting git+git://github.com/czbiohub/xicor/
  Cloning git://github.com/czbiohub/xicor/ to /tmp/pip-req-build-l38weuya
  Running command git clone --filter=blob:none -q
git://github.com/czbiohub/xicor/ /tmp/pip-req-build-l38weuya
  Resolved git://github.com/czbiohub/xicor/ to commit
afe5d368ae974c0237bb670c14ebfc127189327a
  Preparing metadata (setup.py) … done
Requirement already satisfied: scipy>=1.4.1 in
/opt/conda/lib/python3.9/site-packages (from xicor==1.0.1) (1.7.2)
Collecting pytest
  Downloading pytest-6.2.5-py3-none-any.whl (280 kB)
     |                  | 280 kB 15.5 MB/s
Collecting pytest-cov
  Downloading pytest_cov-3.0.0-py3-none-any.whl (20 kB)
Requirement already satisfied: numpy in /opt/conda/lib/python3.9/site-packages
(from xicor==1.0.1) (1.20.3)
Collecting black
  Downloading
black-22.1.0-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.5 MB)
     |                  | 1.5 MB 49.7 MB/s
Collecting pathspec>=0.9.0
  Downloading pathspec-0.9.0-py2.py3-none-any.whl (31 kB)
Collecting tomli>=1.1.0
  Using cached tomli-2.0.0-py3-none-any.whl (12 kB)
Requirement already satisfied: click>=8.0.0 in /opt/conda/lib/python3.9/site-
packages (from black->xicor==1.0.1) (8.0.3)
Requirement already satisfied: typing-extensions>=3.10.0.0 in
/opt/conda/lib/python3.9/site-packages (from black->xicor==1.0.1) (3.10.0.2)
Collecting mypy-extensions>=0.4.3
  Downloading mypy_extensions-0.4.3-py2.py3-none-any.whl (4.5 kB)
Collecting platformdirs>=2
  Downloading platformdirs-2.4.1-py3-none-any.whl (14 kB)
Collecting iniconfig
  Downloading iniconfig-1.1.1-py2.py3-none-any.whl (5.0 kB)
Requirement already satisfied: attrs>=19.2.0 in /opt/conda/lib/python3.9/site-
packages (from pytest->xicor==1.0.1) (21.2.0)
Collecting pluggy<2.0,>=0.12
  Downloading pluggy-1.0.0-py2.py3-none-any.whl (13 kB)
```

```
Collecting py>=1.8.2
  Downloading py-1.11.0-py2.py3-none-any.whl (98 kB)
     |                  | 98 kB 10.0 MB/s
Collecting toml
  Downloading toml-0.10.2-py2.py3-none-any.whl (16 kB)
Requirement already satisfied: packaging in /opt/conda/lib/python3.9/site-
packages (from pytest->xicor==1.0.1) (21.2)
Collecting coverage[toml]>=5.2.1
  Downloading coverage-6.3.1-cp39-cp39-manylinux_2_5_x86_64.manylinux1_x86_64.ma
nylinux_2_17_x86_64.manylinux2014_x86_64.whl (210 kB)
     |                  | 210 kB 48.9 MB/s
Requirement already satisfied: pyparsing<3,>=2.0.2 in
/opt/conda/lib/python3.9/site-packages (from packaging->pytest->xicor==1.0.1)
(2.4.7)
Building wheels for collected packages: xicor
  Building wheel for xicor (setup.py) … done
  Created wheel for xicor: filename=xicor-1.0.1-py3-none-any.whl
size=13979
sha256=cb6f60ad9cd284957d0bd46a4d86c9b7ef42bb1aeebdb1bcff73339884d8f8ed
  Stored in directory: /tmp/pip-ephem-wheel-cache-256e1xv9/wheels/53/f3/b5/52150
b8c6cdfd01e2b78aa76170911584a4e0bfcd2ccdb0928
Successfully built xicor
Installing collected packages: tomli, toml, py, pluggy, iniconfig, coverage,
pytest, platformdirs, pathspec, mypy-extensions, pytest-cov, black, xicor
Successfully installed black-22.1.0 coverage-6.3.1 iniconfig-1.1.1 mypy-
extensions-0.4.3 pathspec-0.9.0 platformdirs-2.4.1 pluggy-1.0.0 py-1.11.0
pytest-6.2.5 pytest-cov-3.0.0 toml-0.10.2 tomli-2.0.0 xicor-1.0.1
```

Import custom modules

[6]:
```
sys.path.append("../python_scripts/")
from featureimportance import (create_simulated_features,
        plot_feature_correlation_spearman,
        calc_new_correlation,
        blr_factor_importance,
        plot_correlationbar,
        rf_factor_importance,
        gradientbars)
```

Settings yaml file

[7]:
```
_fname_settings = 'settings_featureimportance_simulation.yaml'
```

Main function for running the script. Load settings from yaml file

[8]:
```
with open(_fname_settings, 'r') as f:
        settings = yaml.load(f, Loader=yaml.FullLoader)
# Parse settings dictionary as namespace (settings are available as
# settings.variable_name rather than settings['variable_name'])
```

```
settings = SimpleNamespace(**settings)
```

Verify output directory and make it if it does not exist

[9]:
```
os.makedirs(settings.outpath, exist_ok = True)
```

Read data

[10]:
```
data_fieldnames = settings.name_features + [settings.name_target]
df = pd.read_csv(os.path.join(settings.inpath, settings.infname),␣
↪usecols=data_fieldnames)
```

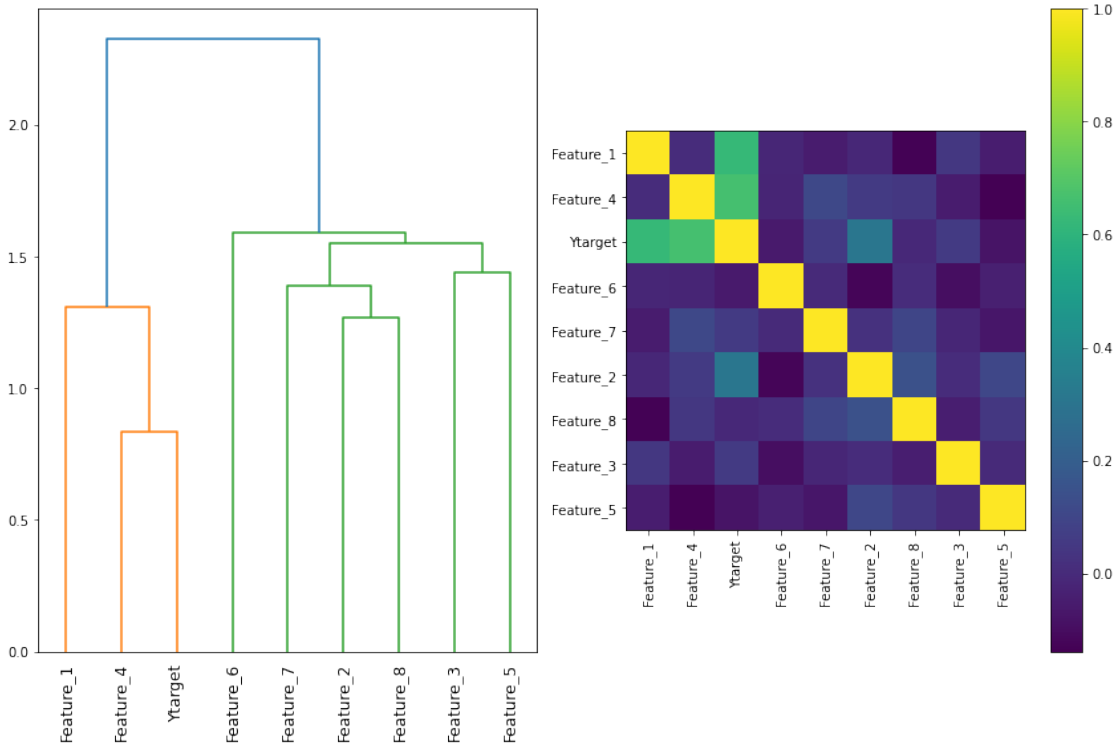
Verify that data is cleaned:

[11]:
```
assert df.select_dtypes(include=['number']).columns.tolist().sort() ==␣
↪data_fieldnames.sort(), 'Data contains non-numeric entries.'
assert df.isnull().sum().sum() == 0, "Data is not cleaned, please run␣
↪preprocess_data.py before"
```

1) Generate Spearman correlation matrix

[12]:
```
print("Calculate Spearman correlation matrix...")
plot_feature_correlation_spearman(df[data_fieldnames].values, data_fieldnames,␣
↪settings.outpath, show = False)
```

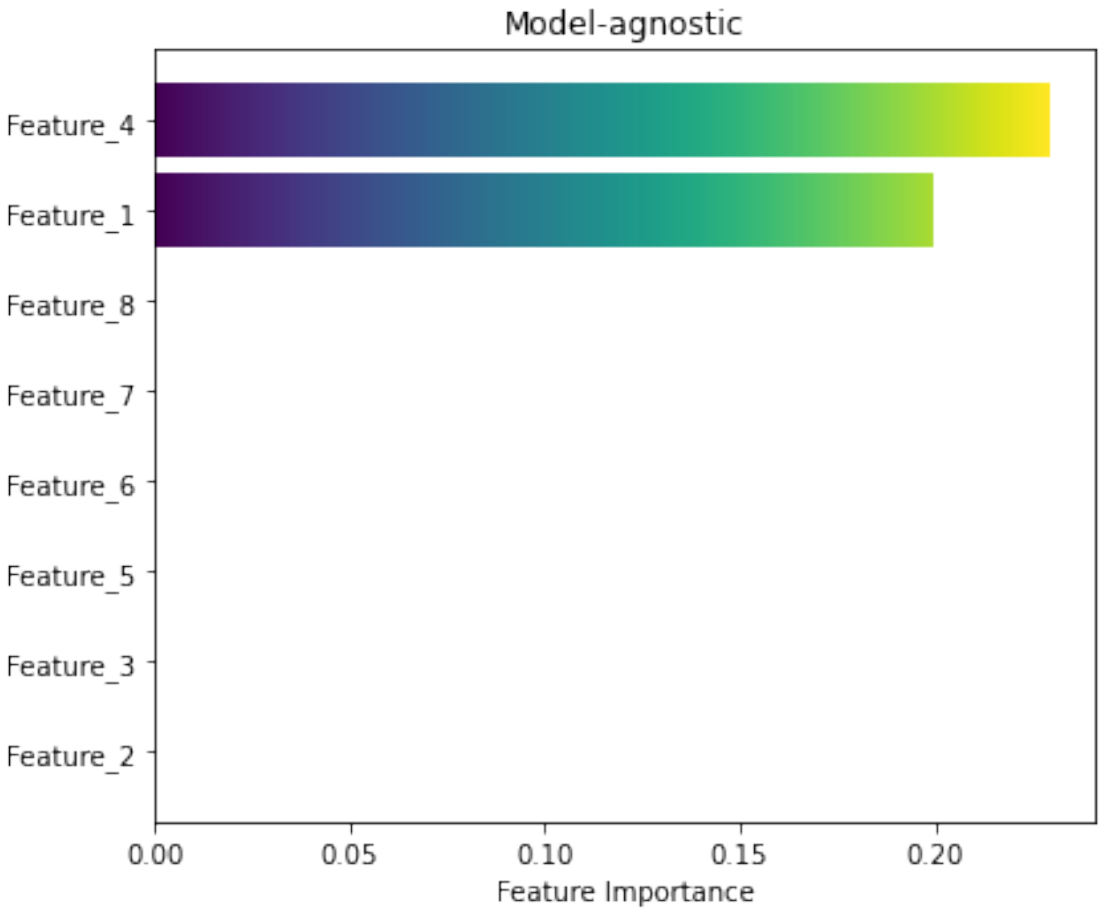
```
Calculate Spearman correlation matrix…
```



2) Generate feature importance based on model-agnostic correlation

[13]:
```
print("Calculate feature importance for mode-agnostic correaltions...")
X = df[settings.name_features].values
y = df[settings.name_target].values
corr = calc_new_correlation(X, y)
plot_correlationbar(corr, settings.name_features, settings.outpath,␣
↪'Model-agnostic-correlation.png', name_method = 'Model-agnostic', show =␣
↪True)
```

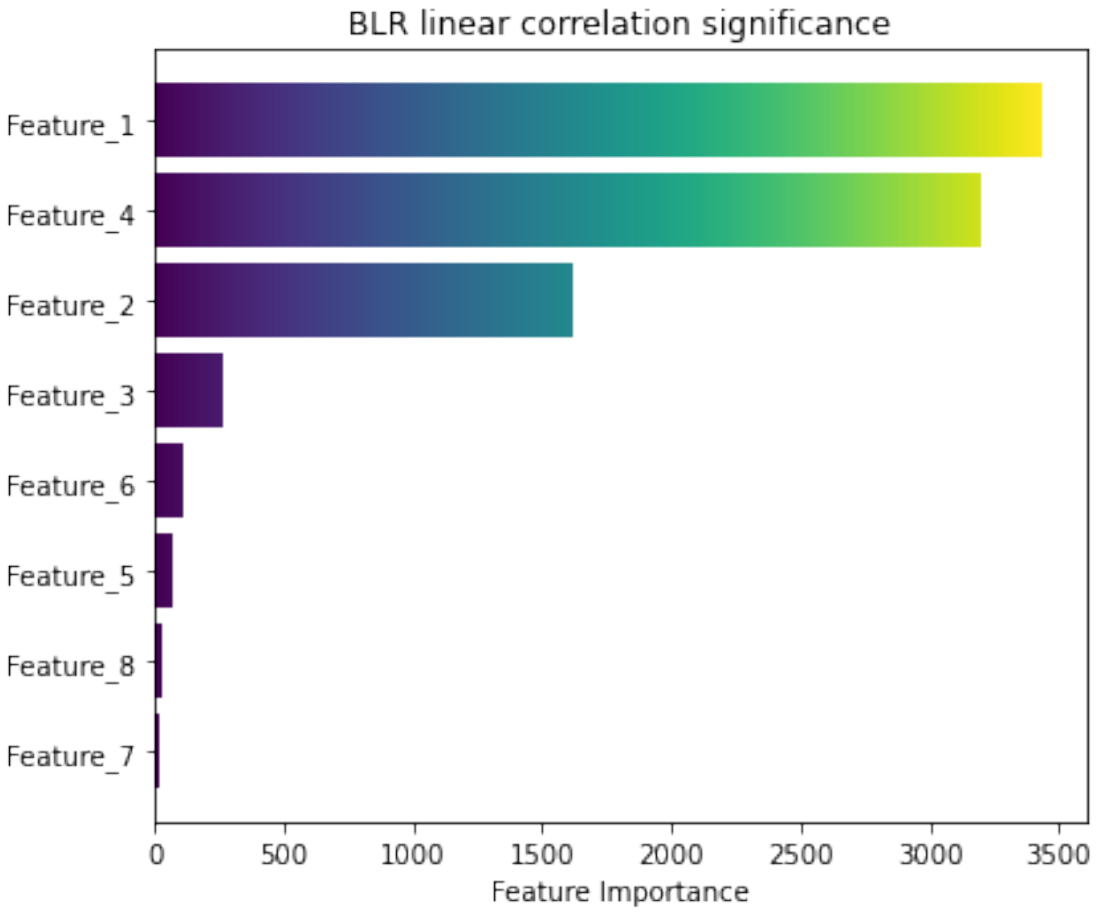
```
Calculate feature importance for mode-agnostic correaltions…

/home/jovyan/workspace/AgReFed-
ML/example_notebooks/../python_scripts/featureimportance.py:350: UserWarning:
Attempting to set identical left == right == 0 results in singular
transformations; automatically expanding.
  ax.imshow(grad, extent=[x,x+w,y,y+h], aspect="auto", zorder=0,
norm=mpl.colors.NoNorm(vmin=0,vmax=1))
```



3) Generate feature importance based on significance of Bayesian Linear Regression coeffcicients:

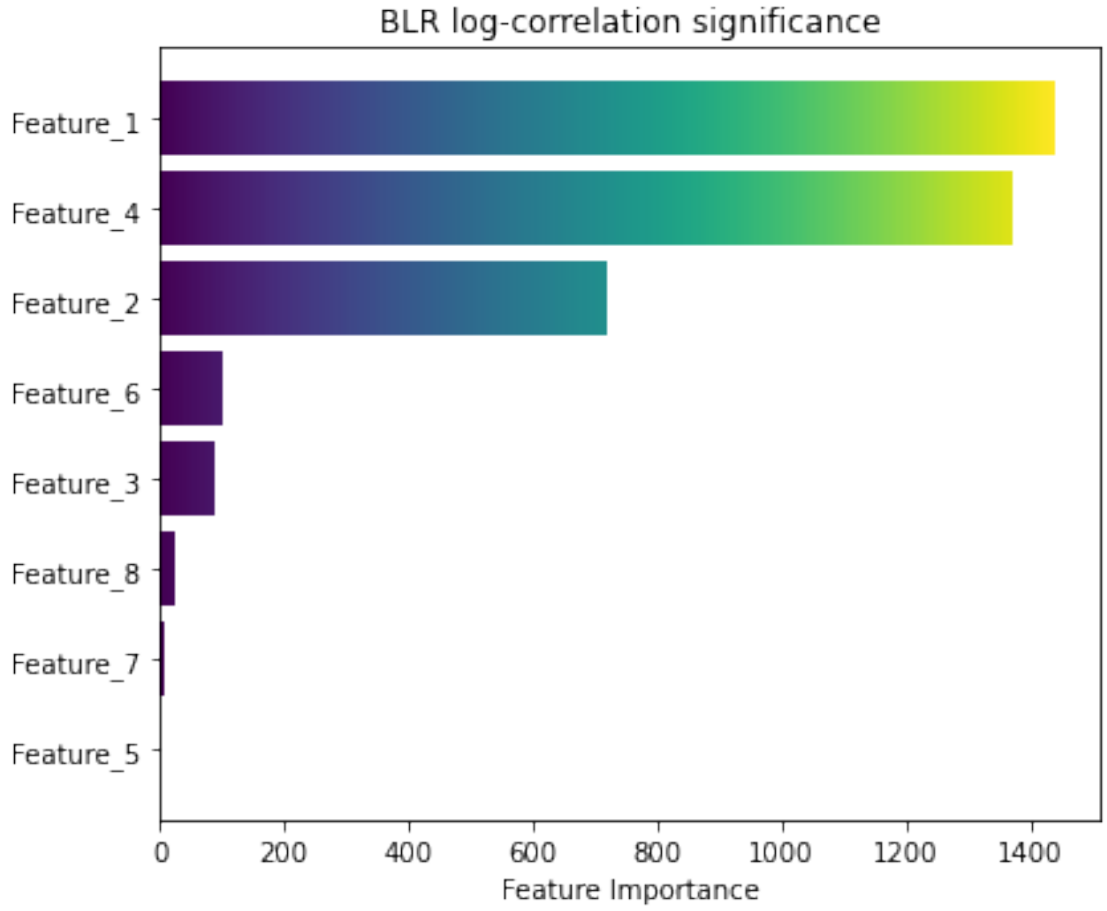
[14]:
```
print("Calculate feature importance for Bayesian Linear Regression...")
corr = blr_factor_importance(X, y, logspace = False)
plot_correlationbar(corr, settings.name_features, settings.outpath,␣
 ↪'BLR-linear-correlation.png', name_method = 'BLR linear correlation␣
 ↪significance', show = True)
# and in log-space
corr = blr_factor_importance(X, y, logspace = True)
plot_correlationbar(corr, settings.name_features, settings.outpath,␣
 ↪'BLR-log-correlation.png', name_method = 'BLR log-correlation significance',␣
 ↪show = True)
```

```
Calculate feature importance for Bayesian Linear Regression...
```




```
/home/jovyan/workspace/AgReFed-
ML/example_notebooks/../python_scripts/featureimportance.py:350: UserWarning:
Attempting to set identical left == right == 0 results in singular
```

```
transformations; automatically expanding.
  ax.imshow(grad, extent=[x,x+w,y,y+h], aspect="auto", zorder=0,
norm=mpl.colors.NoNorm(vmin=0,vmax=1))
```

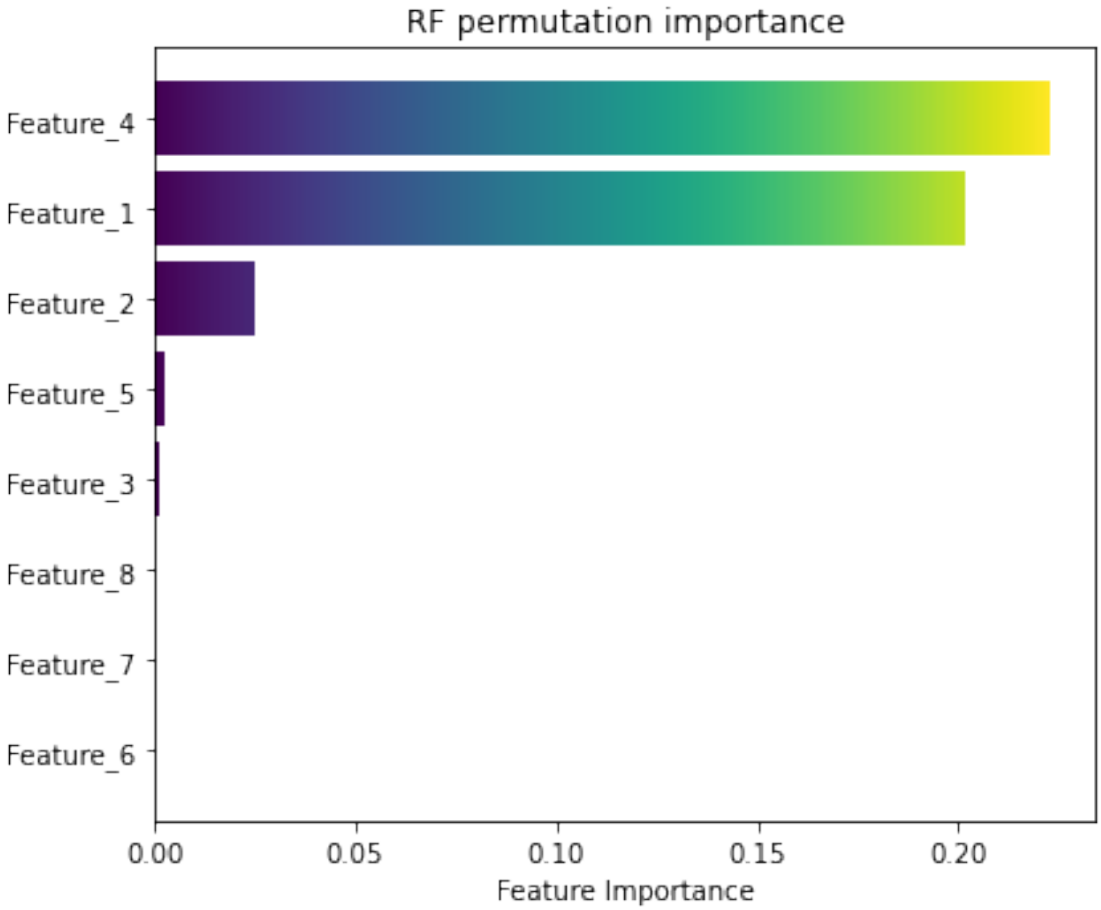


4) Generate feature importance based on Random Forest permutation importance

[15]:
```
print("Calculate feature importance for Random Forest permutation importance...
 ↪")
corr = rf_factor_importance(X, y)
plot_correlationbar(corr, settings.name_features, settings.outpath,␣
 ↪'RF-permutation-importance.png', name_method = 'RF permutation importance',␣
 ↪show = True)
```

```
Calculate feature importance for Random Forest permutation importance…

/home/jovyan/workspace/AgReFed-
ML/example_notebooks/../python_scripts/featureimportance.py:350: UserWarning:
Attempting to set identical left == right == 0 results in singular
transformations; automatically expanding.
  ax.imshow(grad, extent=[x,x+w,y,y+h], aspect="auto", zorder=0,
```

```
norm=mpl.colors.NoNorm(vmin=0,vmax=1))
```



[ ]:

[ ]: