

[Go Up](#)

Name	GaussianProcessRegression
Version	1.0
Description	RFF-accelerated Gaussian Process Regression
License	<a href="http://www.apache.org/licenses/LICENSE-2.0">http://www.apache.org/licenses/LICENSE-2.0</a>
Copyright	Copyright (C) 2022 HPCC Systems®
Authors	HPCCSystems
DependsOn	ML_Core
Platform	8.4.0

## OVERVIEW

### Gaussian Process Regression (GPR)

This bundle provides a Random Fourier Features accelerated version of Gaussian Process Regressor. It allows Data Scientist, researchers or software programmers to apply Gaussian Process Regressor in the parallelized environment of HPCC Systems.

Random Fourier Features(RFF) map the input data to a randomized low-dimensional feature space. Then one can apply fast existing linear methods to such new space and thus accelerate the training of large scale kernel machines[1]. This bundle is the accelerated version of Gaussian Process Regression(GPR) using such random fourier features.

The module GPRI is the main ECL interface. Three functions are available to the users: getSession, fit and predict.

- getSession function generates a 'session ID' for the training and predict process.
- fit function fits the input data and train a GPR model.
- predict function uses the trained GPR model to make predictions for the new observations.

For details of each function, see the comments below above each function in GPRI.ecl file. For details of record structure used in these functions, see Types.ecl file. For usage examples of GPR bundle, see the test cases in Test/test.ecl file.

To use GPR bundle, 'session ID' is required to feed to each fit or predict function call. However, if the training and predict process are in the same session/workunit, getSession only needs to be called once, i.e. fit and predict share same 'session ID' in this case.

## INSTALLATION

Python3 must be installed on each node of HPCC Systems. ML\_Core bundle from [HPCC Systems Machine Learning Library](#) should be installed as well. To install GPR bundle, run following command via HPCC Systems client tool:

```
ecl bundle install https://github.com/hpcc-systems/GaussianProcessRegression.git
```

## EXAMPLES

Test examples are included in the Test folder. Under Test folder, test.ecl file shows the process to define session ID, fit a GPR model and make predictions. The testing data is generated by M\_dataGen.ecl file which generates random test data by user defined size.

## OTHER DOCUMENTATIONS

[HPCC Systems Machine Learning Library](#)

[Using HPCC Systems Machine Learning Library](#)

## ACKNOWLEDGEMENT

This bundle is built upon the original python implementaton of GPR module from below source:  
<https://github.com/gwgundersen/random-fourier-features>

## REFERENCE

[1] Ali Rahimi and Benjamin Recht. 2007. Random features for large-scale kernel machines. In Proceedings of the 20th International Conference on Neural Information Processing Systems (NIPS'07). Curran Associates Inc., Red Hook, NY, USA, 1177–1184.

## Table of Contents

<a href="#">GPRI.ecl</a>
<a href="#">Types.ecl</a>
<a href="#">Internal</a>
<a href="#">Test</a>

# GPRI

---

[Go Up](#)

## IMPORTS

python3 | \_versions.ML\_Core.V3\_2\_2.ML\_Core.Types | std.system.Thorlib | Types |  
Internal.rffGPR |

## DESCRIPTIONS

### GPRI

GPRI
------

No Documentation Found

### Children

1. [GetSession](#) : Initialize GPR on all nodes and return a session ID to be used in the following process
2. [fit](#) : Train a RFF accelerated GPR model
3. [predict](#) : Predict using trained GPR model

---

## GETSESSION

[GPRI](#) /

INTEGER	GetSession
()	

Initialize GPR on all nodes and return a session ID to be used in the following process. This function needs to be called before any other process.

**RETURN** **INTEGER8** — sessID session ID to identify this session.

## FIT

GPRI /

DATASET(Layout_model2)	fit
(INTEGER session, DATASET(NumericField) x, DATASET(NumericField) y, UNSIGNED4 rff_dim = 10, REAL sigma = 1)	

Train a RFF accelerated GPR model

**PARAMETER** **session** ||| INTEGER8 — No Doc

**PARAMETER** **x** ||| TABLE ( NumericField ) — No Doc

**PARAMETER** **y** ||| TABLE ( NumericField ) — No Doc

**PARAMETER** **rff\_dim** ||| UNSIGNED4 — No Doc

**PARAMETER** **sigma** ||| REAL8 — No Doc

**RETURN** **TABLE ( layout\_model2 )** — Gaussian process regression model in Layout\_model2 format.

**PARAMS** session session ID for the training process.

**PARAMS** x independent training data.

**PARAMS** y dependent training data.

**PARAMS** rff\_dim dimension of random fourier features.

**PARAMS** sigma square root of the variance.

**SEE** ML\_Core.Types.Layout\_Model2

## PREDICT

GPRI /

<code>DATASET(NumericField)</code>	<b>predict</b>
<code>(INTEGER session, DATASET(Layout_model2) mod, DATASET(NumericField) x)</code>	

Predict using trained GPR model

**PARAMETER** session ||| INTEGER8 — No Doc

**PARAMETER** mod ||| TABLE ( layout\_model2 ) — No Doc

**PARAMETER** x ||| TABLE ( NumericField ) — No Doc

**RETURN** **TABLE ( NumericField )** — prediction result in NumericField format

**PARAMS** session session ID for the predicting process.

**PARAMS** mod trained GPR model.

**PARAMS** x input data for prediction.

**SEE** ML\_Core.Types.NumericField

---

# Types

---

[Go Up](#)

## DESCRIPTIONS

### TYPES

Types
-------

No Documentation Found

#### Children

1. [initParams](#) : No Documentation Found
- 

### INITPARAMS

[Types](#) /

initParams
------------

No Documentation Found

**FIELD** **nodeid** ||| UNSIGNED4 — No Doc

**FIELD** **nnodes** ||| UNSIGNED4 — No Doc

---

# Internal

---

[Go Up](#)

## Table of Contents

<a href="#">rffGPR.ecl</a>
----------------------------

[Go Up](#)

## IMPORTS

python3 | \_versions.ML\_Core.V3\_2\_2.ML\_Core.Types | std.system.Thorlib | Types |

## DESCRIPTIONS

### **RFFGPR**

	<b>rffGPR</b>
--	---------------

No Documentation Found

### **Children**

1. [init](#): No Documentation Found
2. [fit](#): No Documentation Found
3. [predict](#): No Documentation Found

---

### **INIT**

[rffGPR](#) /



STREAMED DATASET({INTEGER sessID})	<b>init</b>
(STREAMED DATASET(initParams) initDat, STRING wuid = WORKUNIT)	

No Documentation Found

**PARAMETER** **initdat** ||| TABLE ( initParams ) — No Doc

**PARAMETER** **wuid** ||| STRING — No Doc

**RETURN** **TABLE ( { INTEGER8 sessID } )** —

## FIT

rffGPR /

DATASET(Layout_model2)	<b>fit</b>
(INTEGER session, DATASET(NumericField) x, DATASET(NumericField) y, UNSIGNED4 dim = 10, REAL sig = 1)	

No Documentation Found

**PARAMETER** **session** ||| INTEGER8 — No Doc

**PARAMETER** **x** ||| TABLE ( NumericField ) — No Doc

**PARAMETER** **y** ||| TABLE ( NumericField ) — No Doc

**PARAMETER** **dim** ||| UNSIGNED4 — No Doc

**PARAMETER** **sig** ||| REAL8 — No Doc

**RETURN** **TABLE ( layout\_model2 )** —

## PREDICT

rffGPR /

STREAMED DATASET(NumericField)	<b>predict</b>
(STREAMED DATASET(Layout_model2) mod, STREAMED DATASET(NumericField) x, INTEGER session)	

No Documentation Found

**PARAMETER** mod ||| TABLE ( layout\_model2 ) — No Doc

**PARAMETER** x ||| TABLE ( NumericField ) — No Doc

**PARAMETER** session ||| INTEGER8 — No Doc

**RETURN** TABLE ( NumericField ) —

---

# Test

---

[Go Up](#)

## Table of Contents

<a href="#">M_dataGen.ecl</a>
<a href="#">pyGPR.ecl</a>
<a href="#">score.ecl</a>

# Test/ M\_dataGen

---

[Go Up](#)

## IMPORTS

python3 | \_versions.ML\_Core.V3\_2\_2.ML\_Core.Types |

## DESCRIPTIONS

### M\_DATAGEN

a EXPORT	M_dataGen
(INTEGER n, INTEGER n_train)	

No Documentation Found

**PARAMETER** n ||| INTEGER8 — No Doc

**PARAMETER** n\_train ||| INTEGER8 — No Doc

### Children

1. [l](#) : No Documentation Found
2. [toNF](#) : No Documentation Found
3. [generateXData](#) : No Documentation Found
4. [x](#) : No Documentation Found

5. [generateYData](#) : No Documentation Found
  6. [y](#) : No Documentation Found
  7. [X\\_train](#) : No Documentation Found
  8. [Y\\_train](#) : No Documentation Found
  9. [X\\_test](#) : No Documentation Found
  10. [Y\\_test](#) : No Documentation Found
- 

## L

[M\\_dataGen](#) /

<b>l</b>
----------

No Documentation Found

**FIELD** **x** ||| SET ( REAL8 ) — No Doc

---

## TONF

[M\\_dataGen](#) /

<b>toNF</b>
(set of REAL input)

No Documentation Found

**PARAMETER** input ||| SET ( REAL8 ) — No Doc

**RETURN** TABLE ( { UNSIGNED2 wi , UNSIGNED8 id , UNSIGNED4 number , REAL8 value } ) —

---

## GENERATEXDATA

M\_dataGen /

set of real	generateXData
(INTEGER n)	

No Documentation Found

**PARAMETER** n ||| INTEGER8 — No Doc

**RETURN** SET ( REAL8 ) —

---

**X**

M\_dataGen /

x
---

No Documentation Found

---

## GENERATEYDATA

M\_dataGen /

set of real	generateYData
(set of real x_data)	

No Documentation Found

**PARAMETER** x\_data ||| SET ( REAL8 ) — No Doc

**RETURN** SET ( REAL8 ) —

---

**Y**

M\_dataGen /

	<b>y</b>
--	----------

No Documentation Found

---

**X\_TRAIN**

M\_dataGen /

<b>; EXPORT</b>	<b>X_train</b>
-----------------	----------------

No Documentation Found

---

**Y\_TRAIN**

M\_dataGen /

<b>; EXPORT</b>	<b>Y_train</b>
-----------------	----------------

No Documentation Found

---

**X\_TEST**

M\_dataGen /

<b>; EXPORT</b>	<b>X_test</b>
-----------------	---------------

No Documentation Found

---

## Y\_TEST

[M\\_dataGen](#) /

<a href="#">; EXPORT</a>	<a href="#">Y_test</a>
--------------------------	------------------------

No Documentation Found

---



[Go Up](#)

## IMPORTS

python3 | \_versions.ML\_Core.V3\_2\_2.ML\_Core.Types | Types |

## DESCRIPTIONS

### **PYGPR**

<code>DATASET(NumericField)</code>	<b>pyGPR</b>
<code>(DATASET(NumericField) x, DATASET(NumericField) y)</code>	

No Documentation Found

**PARAMETER** **x** ||| TABLE ( NumericField ) — No Doc

**PARAMETER** **y** ||| TABLE ( NumericField ) — No Doc

**RETURN** **TABLE ( NumericField )** —

---

## Test/ score

---

[Go Up](#)

### IMPORTS

```
_versions.ML_Core.V3_2_2.ML_Core | _versions.ML_Core.V3_2_2.ML_Core.Types |  
_versions.PBblas.V3_0_2.PBblas | _versions.PBblas.V3_0_2.PBblas.Types |  
_versions.PBblas.V3_0_2.PBblas.Converted | _versions.PBblas.V3_0_2.PBblas.MatUtils |  
_versions.ML_Core.V3_2_2.ML_Core.Math |
```

### DESCRIPTIONS

#### SCORE

score
(DATASET(NumericField) X=empty_data, DATASET(NumericField) y=empty_data, DATASET(NumericField) Yhat=empty_data)

No Documentation Found

**PARAMETER** x ||| TABLE ( NumericField ) — No Doc

**PARAMETER** y ||| TABLE ( NumericField ) — No Doc

**PARAMETER** yhat ||| TABLE ( NumericField ) — No Doc

**Children**

1. `sumX` : No Documentation Found
  2. `sumy` : No Documentation Found
  3. `sumYY` : No Documentation Found
  4. `n` : No Documentation Found
  5. `x2` : No Documentation Found
  6. `sumX2` : No Documentation Found
  7. `y2` : No Documentation Found
  8. `sumY2` : No Documentation Found
  9. `p1` : No Documentation Found
  10. `p2` : No Documentation Found
  11. `r` : No Documentation Found
  12. `r2` : No Documentation Found
- 

## SUMX

`score` /

	<code>sumX</code>
--	-------------------

No Documentation Found

**RETURN** **REAL8** —

---

## SUMY

`score` /

	<code>sumy</code>
--	-------------------

No Documentation Found

SUMYY

score /

	sumYY
--	-------

No Documentation Found

N

score /

	n
--	---

No Documentation Found

X2

score /

	x2
--	----

No Documentation Found

**RETURN** TABLE ( { UNSIGNED2 wi , UNSIGNED8 id , UNSIGNED4 number , REAL8 value } ) —

---

## SUMX2

score /

	sumX2
--	-------

No Documentation Found

**RETURN** REAL8 —

---

## Y2

score /

	y2
--	----

No Documentation Found

**RETURN** TABLE ( { UNSIGNED2 wi , UNSIGNED8 id , UNSIGNED4 number , REAL8 value } ) —

---

## SUMY2

score /

	sumY2
--	-------

No Documentation Found

P1

score /

p1
----

No Documentation Found

P2

score /

p2
----

No Documentation Found

R

score /

r
---

No Documentation Found

R2

score /

r2
----

No Documentation Found