quant_finance_research_model 1.0

py:Sonar way

xml:Sonar way

2023-03-06

目录

1. quant_finance_research_model Page 1
1.1. 概述 1
1.2. 问题分析 2
1.3. 问题详情 3
1.4. 质量配置 10

1. quant_finance_research_model

报告提供了项目指标的概要，显示了与项目质量相关的最重要的指标。如果需要获取更详细的信息，请登陆网站进一步查询。

报告的项目为quant_finance_research_model，生成时间为2023-03-06，使用的质量配置为py:Sonar way xml:Sonar way，共计 45条规则。

1.1. 概述

编码问题

Bug 0	可靠性修复工作 0min
漏洞 0	安全修复工作 0min
坏味道 176	技术债务 1d0h34min

176 问题	开启问题	176
	重开问题	0
	确认问题	0
	误判问题	0
	不修复的问题	0
	已解决的问题	0
	已删除的问题	0
	阻断	0
	严重	3
	主要	104
	次要	69
	提示	0

静态分析

项目规模

5188
代码行数

行数	6893
方法	444
类	77
文件	58
目录	N/A
重复行(%)	1.5

复杂度

860
复杂度

文件	16.9

注释(%)

11.5
注释(%)

注释行数	675

1.2. 问题分析

违反最多的规则TOP10	
Function names should comply with a naming convention	50
Class names should comply with a naming convention	30
Sections of code should not be commented out	27
Functions, methods and lambdas should not have too many parameters	25
Unused local variables should be removed	19
Local variable and function parameter names should comply with a naming convention	13
"pass" should not be used needlessly	4
Method names should comply with a naming convention	3
Cognitive Complexity of functions should not be too high	3
Nested blocks of code should not be left empty	2

违规最多的文件TOP5	
factor_eval.py	32
tscv.py	11
debug_tscv.py	10
fe_feat.py	10
nn_framework.py	8

复杂度最高的文件TOP5	
factor_eval.py	92
nn_framework.py	87
tscv.py	78
attention.py	51
gbdt_framework.py	49

重复行最多的文件TOP5	
nn_framework.py	34
debug_nn_framework.py	28
factor_eval.py	24
tscv.py	20

1.3. 问题详情

规则	Function names should comply with a naming convention

规则描述	Shared coding conventions allow teams to collaborate efficiently. This rule checks that all function names match a provided regular expression.
	Noncompliant Code Example
	With the default provided regular expression: ^[a-z_][a-z0-9_]{2,30}$
	
	def MyFunction(a,b):
	    ...
	
	Compliant Solution
	
	def my_function(a,b):
	    ...

文件名称	违规行
debug_fe_orth.py	7
catBoost.py	22, 62
lightGBM.py	20, 50
xgBoost.py	38, 69

fe_feat.py	5, 29, 48, 71, 92
debug_factor_eval.py	100, 123
factor_eval.py	17, 23, 29, 37, 45, 53, 65, 105, 119, 133, 148, 162, 176, 187, 198, 218, 258, 264, 270, 289, 298, 311, 324, 333, 349, 365, 381, 397, 413, 424, 435
debug_fe_feat.py	10, 22, 36, 48, 61

规则	Class names should comply with a naming convention

规则描述	Shared coding conventions allow teams to collaborate effectively.
	This rule allows to check that all class names match a provided regular
	expression.
	Noncompliant Code Example
	With default provided regular expression ^[A-Z][a-zA-Z0-9]*$:
	
	class myClass:
	   ...
	
	Compliant Solution
	
	class MyClass:
	   ...

文件名称	违规行
debug_tscv.py	12, 133
wind_cv.py	18, 36, 53
tscv.py	90, 422
debug_gbdt_framework.py	134
basic_embedding.py	6
cls_embedding.py	14
gbdt_framework.py	260
ft_transformer.py	28
debug_nn_framework.py	178
debug_tscv.py	25, 38, 52, 66, 79, 92, 105
tscv.py	331
base_dnn.py	4
nn_framework.py	525
tscv.py	11, 119, 147, 175, 216, 250, 285

规则	Sections of code should not be commented out
规则描述	Programmers should not comment out code as it bloats programs and reduces readability. Unused code should be deleted and can be retrieved from source control history if required. See MISRA C:2004, 2.4 - Sections of code should not be "commented out". MISRA C++:2008, 2-7-2 - Sections of code shall not be "commented out" using C-style comments. MISRA C++:2008, 2-7-3 - Sections of code should not be "commented out" using C++ comments. MISRA C:2012, Dir. 4.4 - Sections of code should not be "commented out"

文件名称	违规行
wind_solve_roll.py	193, 244, 249
debug_eda_feat.py	37
debug_fe_feat.py	89
debug_fe_orth.py	99
debug_fe_protfolio.py	61
debug_fe_val.py	108
debug_tscv.py	146
wind_cv.py	75
wind_get_data.py	39
wind_solve_unroll.py	195
fe_orth.py	32, 48
debug_gbdt_framework.py	140, 148, 197
lightGBM.py	41, 59
debug_fe_val.py	114
debug_factor_eval.py	13, 158
factor_eval.py	142
debug_utils.py	93
debug_fe_utils.py	84
debug_nn_framework.py	232, 235

规则	Functions, methods and lambdas should not have too many parameters

规则描述	A long parameter list can indicate that a new structure should be created to wrap the numerous parameters or that the function is doing too many things.
	Noncompliant Code Example
	With a maximum number of 4 parameters:
	def do_something(param1, param2, param3, param4, param5):
	...
	Compliant Solution
	def do_something(param1, param2, param3, param4):
	...

文件名称	违规行
eda_visualization.py	10, 21, 31, 42, 116, 138
fe_orth.py	63
fe_utils.py	58
cls_embedding.py	41
catBoost.py	22
lightGBM.py	21
xgBoost.py	39
ft_transformer.py	68
attention.py	158
plot_utils.py	5
gbdt_framework.py	171, 217
fe_val.py	133
gbdt_framework.py	29, 107
nn_framework.py	144, 307, 338, 403, 463

规则	Unused local variables should be removed

规则描述	If a local variable is declared but not used, it is dead code and should be removed. Doing so will improve maintainability because developers will
	not wonder what the variable is used for.
	Noncompliant Code Example
	
	def hello(name):
	    message = "Hello " + name # Noncompliant
	    print(name)
	for i in range(10):
	    foo()
	
	Compliant Solution
	
	def hello(name):
	    message = "Hello " + name
	    print(message)
	for _ in range(10):
	    foo()
	
	Exceptions
	_ as well as tuples will not raise an issue for this rule. The following examples are compliant:
	
	for _ in range(10):
	    do_something()
	username, login, password = auth
	do_something_else(username, login)

文件名称	违规行
wind_solve_roll.py	63, 73, 90
debug_fe_orth.py	68, 85
wind_get_data.py	41
wind_solve_unroll.py	53, 62, 77
catBoost.py	51
utils.py	191
debug_factor_eval.py	23
debug_gbdt_framework.py	23
debug_nn_framework.py	25, 193
debug_nn_loss.py	34
utils.py	116
ump_nn.py	144
nn_framework.py	541

规则	Local variable and function parameter names should comply with a naming convention

规则描述

Shared naming conventions allow teams to collaborate effectively. This rule raises an issue when a local variable or function parameter name does
not match the provided regular expression.

Exceptions

Loop counters are ignored by this rule.

```
for i in range(limit):  # Compliant
    print(i)
```

文件名称	违规行
wind_solve_roll.py	108, 190
wind_solve_unroll.py	92
catBoost.py	81, 104
lightGBM.py	66, 88
xgBoost.py	86, 108
fe_feat.py	48, 48, 92, 94

规则	"pass" should not be used needlessly

规则描述

The use of a pass statement where it's not required by the syntax is pure cruft and should be removed.

Noncompliant Code Example

```
def __init__(self, log="", who="", date=0, files=[]):
    self.log = log
    self.files = files
    self.who = who
    self.date = date
    pass    # Noncompliant

def lookup():
    pass    # Compliant; method can't be empty
```

Compliant Solution

```
def __init__(self, log="", who="", date=0, files=[]):
    self.log = log
    self.files = files
    self.who = who
    self.date = date

def lookup():
    pass
```

文件名称	违规行
nn_loss.py	51, 66, 83, 96

规则	Method names should comply with a naming convention

规则描述	Sharing some naming conventions is a key point to make it possible for a team to efficiently collaborate. This rule allows to check that all method
	names match a provided regular expression.
	Noncompliant Code Example
	With default provided regular expression: ^[a-z_][a-z0-9_]{2,30}$
	
	class MyClass:
	    def MyMethod(a,b):
	        ...
	
	Compliant Solution
	
	class MyClass:
	    def my_method(a,b):
	        ...

文件名称	违规行
gbdt_framework.py	171, 217
nn_framework.py	403

规则	Cognitive Complexity of functions should not be too high
规则描述	Cognitive Complexity is a measure of how hard the control flow of a function is to understand. Functions with high Cognitive Complexity will be difficult to maintain. See Cognitive Complexity

文件名称	违规行
attention.py	158
utils.py	62
ump_nn.py	37

规则	Nested blocks of code should not be left empty
规则描述	Most of the time a block of code is empty when a piece of code is really missing. So such empty block must be either filled or removed. Noncompliant Code Example for i in range(3): pass Exceptions When a block contains a comment, this block is not considered to be empty.

文件名称	违规行
tscv.py	488

fe_feat.py	37

1.4. 质量配置

质量配置	py:Sonar way Bug:10 坏味道:21

规则	类型	违规级别
"yield" and "return" should not be used outside functions	Bug	阻断
"return" and "yield" should not be used in the same function	Bug	阻断
"__exit__" should accept type, value, and traceback arguments	Bug	阻断
"__init__" should not return a value	Bug	阻断
"break" and "continue" should not be used outside a loop	Bug	严重
Variables should not be self-assigned	Bug	主要
Related "if/else if" statements should not have the same condition	Bug	主要
Increment and decrement operators should not be used	Bug	主要
Identical expressions should not be used on both sides of a binary operator	Bug	主要
Jump statements should not be followed by dead code	Bug	主要
Backticks should not be used	坏味道	阻断
The "exec" statement should not be used	坏味道	阻断
Methods and field names should not differ only by capitalization	坏味道	阻断
Cognitive Complexity of functions should not be too high	坏味道	严重
Track uses of "FIXME" tags	坏味道	主要
Functions, methods and lambdas should not have too many parameters	坏味道	主要
Nested blocks of code should not be left empty	坏味道	主要
Collapsible "if" statements should be merged	坏味道	主要
Function names should comply with a naming convention	坏味道	主要
A field should not duplicate the name of its containing class	坏味道	主要
Two branches in a conditional structure should not have exactly the same implementation	坏味道	主要
Redundant pairs of parentheses should be removed	坏味道	主要
The "print" statement should not be used	坏味道	主要
Sections of code should not be commented out	坏味道	主要
"<>" should not be used to test inequality	坏味道	主要
Field names should comply with a naming convention	坏味道	次要

Local variable and function parameter names should comply with a naming convention	坏味道	次要
"pass" should not be used needlessly	坏味道	次要
Method names should comply with a naming convention	坏味道	次要
Class names should comply with a naming convention	坏味道	次要
Unused local variables should be removed	坏味道	次要

质量配置	xml:Sonar way Bug:1 坏味道:3	
规则	类型	违规级别
XML files containing a prolog header should start with "<?xml" characters	Bug	严重
Track uses of "FIXME" tags	坏味道	主要
Sections of code should not be commented out	坏味道	主要
Track uses of "TODO" tags	坏味道	提示