

BotSearcher

Создано системой Doxygen 1.8.18

1	Алфавитный указатель пространств имен	1
1.1	Пространства имен	1
2	Иерархический список классов	3
2.1	Иерархия классов	3
3	Алфавитный указатель классов	5
3.1	Классы	5
4	Список файлов	7
4.1	Файлы	7
5	Пространства имен	9
5.1	Пространство имен BinaryPlot	9
5.1.1	Функции	9
5.1.1.1	drow_3d_plot()	9
5.1.1.2	plot_local_outlier_factor()	10
5.2	Пространство имен BotFinderConfigs	10
5.2.1	Переменные	11
5.2.1.1	database	11
5.2.1.2	host	11
5.2.1.3	password	11
5.3	Пространство имен main	11
5.3.1	Переменные	11
5.3.1.1	bf	12
5.3.1.2	start_status	12
5.4	Пространство имен Samples	12
5.5	Пространство имен UserDataBase	12
5.6	Пространство имен VPTree	12
6	Классы	13
6.1	Класс main.BotFinder	13
6.1.1	Подробное описание	14
6.1.2	Конструктор(ы)	14
6.1.2.1	__init__()	14
6.1.3	Методы	14
6.1.3.1	is_first_start()	14
6.1.3.2	load_data()	14
6.1.3.3	load_list_of_player()	15
6.1.3.4	load_samples_from_config_file()	15
6.1.3.5	make_report()	15
6.1.3.6	prepare_work_place()	15
6.1.3.7	try_load_data_base()	16
6.1.4	Данные класса	16
6.1.4.1	data_base	16

6.1.4.2 main_config	16
6.1.4.3 samples_config	16
6.1.4.4 shema	16
6.2 Класс BotFinderConfigs.DataBaseConfig	17
6.2.1 Подробное описание	17
6.2.2 Конструктор(ы)	18
6.2.2.1 __init__()	18
6.2.3 Методы	18
6.2.3.1 default_path()	18
6.2.3.2 load_or_create()	18
6.2.4 Данные класса	18
6.2.4.1 database	18
6.2.4.2 host	19
6.2.4.3 password	19
6.2.4.4 user	19
6.3 Класс main.HTMLreport	19
6.3.1 Подробное описание	20
6.3.2 Конструктор(ы)	20
6.3.2.1 __init__()	20
6.3.3 Методы	20
6.3.3.1 make_report()	20
6.3.3.2 pacage_bloc()	21
6.3.3.3 package_img_bloc()	21
6.3.3.4 package_img_bloc_tuple()	22
6.3.3.5 package_table()	22
6.3.3.6 package_title()	22
6.3.3.7 prepare_all_image()	23
6.3.3.8 prepare_dir()	23
6.3.4 Данные класса	23
6.3.4.1 path_dir	23
6.4 Класс BotFinderConfigs.MainConfig	24
6.4.1 Подробное описание	25
6.4.2 Конструктор(ы)	25
6.4.2.1 __init__()	25
6.4.3 Методы	25
6.4.3.1 default_path()	25
6.4.4 Данные класса	25
6.4.4.1 count_neighbors	25
6.4.4.2 eject_lip	26
6.4.4.3 ignore_error_with_session_element_missing	26
6.4.4.4 parse_samples_from_data_base	26
6.4.4.5 refresh_player_on_load	26
6.4.4.6 refresh_samples_on_load	26

6.4.4.7 refresh_single_samples_element_if_missing	27
6.4.4.8 report_folder_name	27
6.4.4.9 samples_folder_name	27
6.4.4.10 samples_player_name_file	27
6.4.4.11 save_data_base_data	27
6.4.4.12 save_data_base_passwor	28
6.4.4.13 save_session_data	28
6.5 Класс BotFinderConfigs.ProgramConfig	28
6.5.1 Подробное описание	28
6.5.2 Методы	29
6.5.2.1 config_file_is_available()	29
6.5.2.2 default_path()	30
6.5.2.3 load_or_create()	30
6.6 Класс Samples.Sample	31
6.6.1 Подробное описание	31
6.6.2 Конструктор(ы)	32
6.6.2.1 __init__()	32
6.6.3 Методы	32
6.6.3.1 append_player_from_data_base()	32
6.6.3.2 export_to_file()	32
6.6.3.3 import_from_file()	33
6.6.3.4 interquartile_range()	33
6.6.3.5 linear_ejection()	33
6.6.3.6 median()	34
6.6.3.7 quartile()	34
6.6.3.8 remissible_segment()	34
6.6.3.9 try_recover_from_data_base()	35
6.6.3.10 verified_segment()	35
6.6.4 Данные класса	35
6.6.4.1 data_base_request	35
6.6.4.2 max	35
6.6.4.3 min	35
6.6.4.4 name	36
6.7 Класс BotFinderConfigs.SamplesConfig	36
6.7.1 Подробное описание	36
6.7.2 Конструктор(ы)	37
6.7.2.1 __init__()	37
6.7.3 Методы	37
6.7.3.1 default_path()	37
6.7.3.2 default_sample()	37
6.7.4 Данные класса	37
6.7.4.1 sample_model	37
6.7.4.2 samples_configs	38

6.7.4.3	<code>user_query_from_data_base</code>	38
6.8	Класс <code>Samples.SchemaPlayerSamples</code>	38
6.8.1	Подробное описание	39
6.8.2	Конструктор(ы)	39
6.8.2.1	<code>__init__()</code>	39
6.8.3	Методы	39
6.8.3.1	<code>__getitem__()</code>	40
6.8.3.2	<code>append_sample()</code>	40
6.8.3.3	<code>clear_from_incomplete()</code>	40
6.8.3.4	<code>distance()</code>	41
6.8.3.5	<code>local_outlier_factor()</code>	41
6.8.3.6	<code>local_reachability_density()</code>	42
6.8.3.7	<code>nearest_distance()</code>	42
6.8.3.8	<code>outliers_point()</code>	43
6.8.3.9	<code>point_on_nearest_distance()</code>	43
6.8.3.10	<code>reachability_distance()</code>	43
6.8.4	Данные класса	44
6.8.4.1	<code>lip</code>	44
6.8.4.2	<code>neighbour_count</code>	44
6.8.4.3	<code>points</code>	44
6.8.4.4	<code>samples</code>	44
6.8.4.5	<code>vp_tree</code>	45
6.8.4.6	<code>VPtree</code>	45
6.9	Класс <code>VPtree.VPtree</code>	45
6.9.1	Подробное описание	46
6.9.2	Конструктор(ы)	46
6.9.2.1	<code>__init__()</code>	46
6.9.3	Методы	46
6.9.3.1	<code>get_n_nearest_neighbors()</code>	46
6.9.4	Данные класса	46
6.9.4.1	<code>distance</code>	47
6.9.4.2	<code>left</code>	47
6.9.4.3	<code>left_max</code>	47
6.9.4.4	<code>left_min</code>	47
6.9.4.5	<code>right</code>	47
6.9.4.6	<code>right_max</code>	48
6.9.4.7	<code>right_min</code>	48
6.9.4.8	<code>vantage_point</code>	48
7	Файлы	49
7.1	Файл <code>BinaryPlot.py</code>	49
7.2	Файл <code>BotFinderConfigs.py</code>	49
7.3	Файл <code>main.py</code>	50

7.4 Файл Samples.py	50
7.5 Файл UserDataBase.py	50
7.6 Файл VPTree.py	50
Предметный указатель	51

Глава 1

Алфавитный указатель пространств имен

1.1 Пространства имен

Полный список пространств имен.

BinaryPlot	9
BotFinderConfigs	10
main	11
Samples	12
UserDataBase	12
VPTree	12

Глава 2

Иерархический список классов

2.1 Иерархия классов

Иерархия классов.

main.BotFinder	13
dict	
Samples.Sample	31
main.HTMLreport	19
Samples.ShemaPlayerSamples	38
VPTree.VPTree	45
ABC	
BotFinderConfigs.ProgramConfig	28
BotFinderConfigs.DataBaseConfig	17
BotFinderConfigs.MainConfig	24
BotFinderConfigs.SamplesConfig	36

Глава 3

Алфавитный указатель классов

3.1 Классы

Классы с их кратким описанием.

main.BotFinder	Основной класс программы, реализующий запуск чтения/генерации конфигов программы, функции анализа данных и формирования отчетов	13
BotFinderConfigs.DataBaseConfig	загрузка файла конфигурации базы данных с диалоговыми опциями	17
main.HTMLreport	класс формирования html отчета	19
BotFinderConfigs.MainConfig	класс основной конфигурации программы	24
BotFinderConfigs.ProgramConfig	Абстрактный класс конфигурационных файлов программы	28
Samples.Sample	Линейная Выборка данных	31
BotFinderConfigs.SamplesConfig	Конфигурация выборки	36
Samples.ShemaPlayerSamples	класс для комплексного анализа одномерных выборок	38
VPTree.VPTree	45

Глава 4

Список файлов

4.1 Файлы

Полный список файлов.

BinaryPlot.py	49
BotFinderConfigs.py	49
main.py	50
Samples.py	50
UserDataBase.py	50
VPTree.py	50

Глава 5

Пространства имен

5.1 Пространство имен BinaryPlot

Функции

- def `plot_local_outlier_factor` (names, segments, corrupt_points, clear_points)
построение графика визуально отображающего методов локального выброса и метода тьюки для двух измерений
- def `drow_3d_plot` (names, segments, corrupt_points, clear_points)
построение графика визуально отображающего методов локального выброса для трех измерений

5.1.1 Функции

5.1.1.1 `drow_3d_plot()`

```
def BinaryPlot.drow_3d_plot (  
    names,  
    segments,  
    corrupt_points,  
    clear_points )
```

построение графика визуально отображающего методов локального выброса для трех измерений

Аргументы

names	название выборок
segments	границы графика
corrupt_points	точки выброса
clear_points	нормальные точки

Возвращает

фигура matplotlib с нарисованным графиком

См. определение в файле BinaryPlot.py строка 61

5.1.1.2 plot_local_outlier_factor()

```
def BinaryPlot.plot_local_outlier_factor (
    names,
    segments,
    corrupt_points,
    clear_points )
```

построение графика визуально отображающего методов локального выброса и метода тыюки для двух измерений

Аргументы

names	название выборок
segments	границы графика
corrupt_points	значения выброса
clear_points	нормальные значение

Возвращает

фигура matplotlib с нарисованным графиком

См. определение в файле BinaryPlot.py строка 16

5.2 Пространство имен BotFinderConfigs

Классы

- class [DataBaseConfig](#)
загрузка файла конфигурации базы данных с диалоговыми опциями
- class [MainConfig](#)
класс основной конфигурации программы
- class [ProgramConfig](#)
Абстрактный класс конфигурационных файлов программы
- class [SamplesConfig](#)
Конфигурация выборки

Переменные

- [host](#)
- [database](#)
- [password](#)

5.2.1 Переменные

5.2.1.1 database

BotFinderConfigs.database

См. определение в файле BotFinderConfigs.py строка 169

5.2.1.2 host

BotFinderConfigs.host

См. определение в файле BotFinderConfigs.py строка 168

5.2.1.3 password

BotFinderConfigs.password

См. определение в файле BotFinderConfigs.py строка 177

5.3 Пространство имен main

Классы

- class [BotFinder](#)
Основной класс программы, реализующий запуск чтения/генерации конфигов программы, функции анализа данных и формирования отчетов
- class [HTMLreport](#)
класс формирования html отчета

Переменные

- [start_status](#) = [BotFinder.is_first_start\(\)](#)
- [bf](#) = [BotFinder\(\)](#)

5.3.1 Переменные

5.3.1.1 bf

```
main.bf = BotFinder()
```

См. определение в файле main.py строка 338

5.3.1.2 start_status

```
main.start_status = BotFinder.is\_first\_start()
```

См. определение в файле main.py строка 337

5.4 Пространство имен Samples

Классы

- class [Sample](#)
 Линейная Выборка данных
- class [ShemaPlayerSamples](#)
 класс для комплексного анализа одномерных выборок

5.5 Пространство имен UserDataBase

5.6 Пространство имен VPTree

Классы

- class [VPTree](#)

Глава 6

Классы

6.1 Класс main.BotFinder

Основной класс программы, реализующий запуск чтения/генерации конфигов программы, функции анализа данных и формирования отчетов

Открытые члены

- `def __init__ (self)`
- `def load_samples_from_config_file (self)`
загрузить данные о выборках из конфигурации
- `def try_load_data_base (self)`
- `def prepare_work_place (self)`
Подготовить файлы выборок и дерикотрию для их хранения.
- `def load_list_of_player (self)`
Загрузить список пользователей из файла или базы данных
- `def load_data (self)`
Загрузить выборки из файла или из базы данных.
- `def make_report (self)`
Подготовить данные для создания отчета и сохранить отчет в папке

Открытые статические члены

- `def is_first_start ()`
проверка на первый запуск программы.

Открытые атрибуты

- `main_config`
оснвая конфигурация программы
- `data_base`
объект подключение к базе данных
- `samples_config`
конфигурация выборок
- `shema`
набор одномерных выборок

6.1.1 Подробное описание

Основной класс программы, реализующий запуск чтения/генерации кофигов программы, функции анализа данных и формирования отчетов

См. определение в файле main.py строка 182

6.1.2 Конструктор(ы)

6.1.2.1 `__init__()`

```
def main.BotFinder.__init__(  
    self )
```

См. определение в файле main.py строка 184

6.1.3 Методы

6.1.3.1 `is_first_start()`

```
def main.BotFinder.is_first_start ( ) [static]
```

проверка на первый запуск программы.

Если в папке с программой отсутствует основной конфигурационный файл она считается впервые запущенной

Returns True если программа запущена впервые иначе false

См. определение в файле main.py строка 205

6.1.3.2 `load_data()`

```
def main.BotFinder.load_data (  
    self )
```

Загрузить выборки из файла или из базы данных.

После загрузки проверить данные на совместимость загрузить их в файл

См. определение в файле main.py строка 279

6.1.3.3 load_list_of_player()

```
def main.BotFinder.load_list_of_player (
    self )
```

Загрузить список пользователей из файла или базы данных

Возвращает

список идентификаторов игроков

См. определение в файле main.py строка 262

6.1.3.4 load_samples_from_config_file()

```
def main.BotFinder.load_samples_from_config_file (
    self )
```

загрузить данные о выборках из конфигурации

Аргументы

cfg	конфигурация выборок
-----	----------------------

См. определение в файле main.py строка 215

6.1.3.5 make_report()

```
def main.BotFinder.make_report (
    self )
```

Подготовить данные для создания отчета и сохранить отчет в папке

См. определение в файле main.py строка 312

6.1.3.6 prepare_work_place()

```
def main.BotFinder.prepare_work_place (
    self )
```

Подготовить файлы выборок и дерикотрию для их хранения.

См. определение в файле main.py строка 242

6.1.3.7 try_load_data_base()

```
def main.BotFinder.try_load_data_base (
    self )
```

См. определение в файле main.py строка 224

6.1.4 Данные класса

6.1.4.1 data_base

```
main.BotFinder.data_base
```

объект подключение к базе данных

См. определение в файле main.py строка 189

6.1.4.2 main_config

```
main.BotFinder.main_config
```

оснвая конфигурация программы

См. определение в файле main.py строка 187

6.1.4.3 samples_config

```
main.BotFinder.samples_config
```

конфигурация выборок

См. определение в файле main.py строка 193

6.1.4.4 shema

```
main.BotFinder.shema
```

набор одномерных выборок

См. определение в файле main.py строка 195

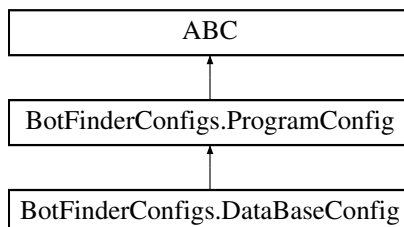
Объявления и описания членов класса находятся в файле:

- [main.py](#)

6.2 Класс BotFinderConfigs.DataBaseConfig

загрузка файла конфигурации базы данных с диалоговыми опциями

Граф наследования: BotFinderConfigs.DataBaseConfig:



Открытые члены

- `def __init__ (self)`
- `def default_path (self)`
Путь по умолчанию
- `def load_or_create (self, path=None, save_data=True, save_password=False)`

Открытые атрибуты

- `host`
адрес базы данных
- `database`
имя базы данных
- `user`
пользователь базы данных
- `password`
пароль базы данных

6.2.1 Подробное описание

загрузка файла конфигурации базы данных с диалоговыми опциями

Аргументы

<code>path</code>	путь к файлу конфигурации
<code>save_data</code>	сохранять ли данные подключения
<code>save_password</code>	сохранять ли пароль

Возвращает

объект загруженной конфигурации

См. определение в файле BotFinderConfigs.py строка 119

6.2.2 Конструктор(ы)

6.2.2.1 `__init__()`

```
def BotFinderConfigs.DataBaseConfig.__init__(  
    self )
```

См. определение в файле BotFinderConfigs.py строка 122

6.2.3 Методы

6.2.3.1 `default_path()`

```
def BotFinderConfigs.DataBaseConfig.default_path(  
    self )
```

Путь по умолчанию

Переопределяет метод предка [BotFinderConfigs.ProgramConfig](#).

См. определение в файле BotFinderConfigs.py строка 140

6.2.3.2 `load_or_create()`

```
def BotFinderConfigs.DataBaseConfig.load_or_create(  
    self,  
    path = None,  
    save_data = True,  
    save_password = False )
```

См. определение в файле BotFinderConfigs.py строка 143

6.2.4 Данные класса

6.2.4.1 `database`

BotFinderConfigs.DataBaseConfig.database

имя базы данных

См. определение в файле BotFinderConfigs.py строка 127

6.2.4.2 host

BotFinderConfigs.DataBaseConfig.host

адрес базы данных

См. определение в файле BotFinderConfigs.py строка 124

6.2.4.3 password

BotFinderConfigs.DataBaseConfig.password

пароль базы данных

См. определение в файле BotFinderConfigs.py строка 136

6.2.4.4 user

BotFinderConfigs.DataBaseConfig.user

пользователь базы данных

диалоговый ввод всех данных подключения

См. определение в файле BotFinderConfigs.py строка 133

Объявления и описания членов класса находятся в файле:

- [BotFinderConfigs.py](#)

6.3 Класс main.HTMLreport

класс формирования html отчета

Открытые члены

- def `__init__` (self, str path)
- def `prepare_dir` (self)
 - Подготовить папку для хранения отчета, в случае если папка отчета уже существует, удалить её и создать новую.
- def `make_report` (self, samples_lable, ejection_data, other_data, color_other_data)
 - Сформировать содержание html отчета
- def `package_title` (self, title)
 - Упаковать заголовок в html контейнер, с ориентированием по центру
- def `prepare_all_image` (self, names, segments, corrupt_points, clear_points)
 - Нарисовать графики и сохранить их в папке с отчетом
- def `package_img_bloc` (self, picture_number)
 - Упаковать картинку из папки отчета в контейнер, с ориентированием по центру
- def `pacage_bloc` (self, title, content)
 - Создать html код блока с заголовком
- def `package_img_bloc_tuple` (self, picture_number_one, picture_number_two)
 - Упаковать две картинки из папки отчета в контейнер в один блок
- def `package_table` (self, title, titles_column, data, color=None)
 - Сформировать html код таблицы данных

Открытые атрибуты

- `path_dir`
путь к папке хранения отчета

6.3.1 Подробное описание

класс формирования html отчета

См. определение в файле main.py строка 13

6.3.2 Конструктор(ы)

6.3.2.1 `__init__()`

```
def main.HTMLreport.__init__(
    self,
    str path )
```

Аргументы

path	путь к папке хранения отчета
------	------------------------------

См. определение в файле main.py строка 19

6.3.3 Методы

6.3.3.1 `make_report()`

```
def main.HTMLreport.make_report (
    self,
    samples_lable,
    ejection_data,
    other_data,
    color_other_data )
```

Сформировать содержание html отчета

Аргументы

samples_lable	название выборок
ejection_data	данные точек выброса
other_data	данные точек выброса линейного анализа
color_other_data	цвета таблицы точек выброса линейного анализа текст html отчета

См. определение в файле main.py строка 45

6.3.3.2 package_bloc()

```
def main.HTMLreport.package_bloc (
    self,
    title,
    content )
```

Создать html код блока с заголовком

Аргументы

title	заголовок блока
content	содержание блока

Возвращает

html код

См. определение в файле main.py строка 139

6.3.3.3 package_img_bloc()

```
def main.HTMLreport.package_img_bloc (
    self,
    picture_number )
```

Упаковать картинку из папки отчета в контейнер, с ориентированием по центру

Аргументы

picture_number	номер картинки
----------------	----------------

Возвращает

html код

См. определение в файле main.py строка 127

6.3.3.4 package_img_bloc_tuple()

```
def main.HTMLreport.package_img_bloc_tuple (
    self,
    picture_number_one,
    picture_number_two )
```

Упаковать две картинки из папки отчета в контейнер в один блок

picture_number номер картинки

Возвращает

html код

См. определение в файле main.py строка 149

6.3.3.5 package_table()

```
def main.HTMLreport.package_table (
    self,
    title,
    titles_column,
    data,
    color = None )
```

Сформировать html код таблицы данных

title заголовок таблицы titles_column название колонок таблицы data содержание таблицы color цвета строк таблицы

Возвращает

html код

См. определение в файле main.py строка 162

6.3.3.6 package_title()

```
def main.HTMLreport.package_title (
    self,
    title )
```

Упаковать заголовок в html контейнер, с ориентированием по центру

Аргументы

title	заголовок html код
-------	--------------------

См. определение в файле main.py строка 81

6.3.3.7 prepare_all_image()

```
def main.HTMLreport.prepare_all_image (
    self,
    names,
    segments,
    corrupt_points,
    clear_points )
```

Нарисовать графики и сохранить их в папке с отчетом

Аргументы

names	названия выборок
segments	отрезок для увеличенного графика для каждой выборки
corrupt_points	точки выброса данных
clear_points	точки нормальных данных

См. определение в файле main.py строка 93

6.3.3.8 prepare_dir()

```
def main.HTMLreport.prepare_dir (
    self )
```

Подготовить папку для хранения отчета, в случае если папка отчета уже существует, удалить её и создать новую.

См. определение в файле main.py строка 30

6.3.4 Данные класса

6.3.4.1 path_dir

main.HTMLreport.path_dir

путь к папке хранения отчета

См. определение в файле main.py строка 21

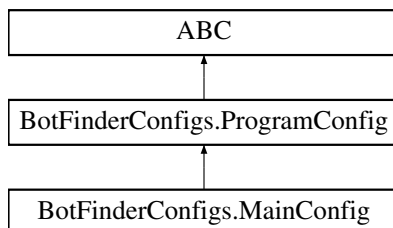
Объявления и описания членов класса находятся в файле:

- [main.py](#)

6.4 Класс BotFinderConfigs.MainConfig

класс основной конфигурации программы

Граф наследования: BotFinderConfigs.MainConfig:



Открытые члены

- `def __init__ (self)`
- `def default_path (self)`
Путь по умолчанию

Открытые атрибуты

- `parse_samples_from_data_base`
использовать базу данных для получения выборок
- `save_data_base_data`
сохранить данные для входа в базу данных
- `save_data_base_passwor`
- `refresh_samples_on_load`
пересоздавать данные выборки из базы данных при каждом запуске программы
- `refresh_player_on_load`
пересоздавать данные выборки пользователей из базы данных при каждом запуске программы
- `refresh_single_samples_element_if_missing`
заново загрузить элемент выборки, если он не будет найден в файлах
- `ignore_error_with_session_element_missing`
игнорировать ошибки связанные с отсутствием элемента выборки
- `save_session_data`
сохранять выборки в файлах
- `samples_folder_name`
Название папки для хранения выборок
- `samples_player_name_file`
Название файла для хранения выборки игроков
- `report_folder_name`
название папки для хранения отчета
- `count_neighbors`
количество соседей в методе локального уровня вброса
- `eject_lip`
порог выброса в методе локального уровня вброса

6.4.1 Подробное описание

класс основной конфигурации программы

См. определение в файле BotFinderConfigs.py строка 56

6.4.2 Конструктор(ы)

6.4.2.1 `__init__()`

```
def BotFinderConfigs.MainConfig.__init__(  
    self )
```

См. определение в файле BotFinderConfigs.py строка 57

6.4.3 Методы

6.4.3.1 `default_path()`

```
def BotFinderConfigs.MainConfig.default_path(  
    self )
```

Путь по умолчанию

Переопределяет метод предка [BotFinderConfigs.ProgramConfig](#).

См. определение в файле BotFinderConfigs.py строка 105

6.4.4 Данные класса

6.4.4.1 `count_neighbors`

BotFinderConfigs.MainConfig.count_neighbors

количество соседей в методе локального уровня вброса

См. определение в файле BotFinderConfigs.py строка 98

6.4.4.2 eject_lip

BotFinderConfigs.MainConfig.eject_lip

порог выброса в методе локального уровня вброса

См. определение в файле BotFinderConfigs.py строка 101

6.4.4.3 ignore_error_with_session_element_missing

BotFinderConfigs.MainConfig.ignore_error_with_session_element_missing

игнорировать ошибки связанные с отсутствием элемента выборки

См. определение в файле BotFinderConfigs.py строка 80

6.4.4.4 parse_samples_from_data_base

BotFinderConfigs.MainConfig.parse_samples_from_data_base

использовать базу данных для получения выборок

См. определение в файле BotFinderConfigs.py строка 59

6.4.4.5 refresh_player_on_load

BotFinderConfigs.MainConfig.refresh_player_on_load

пересоздавать данные выборки пользователей из базы данных при каждом запуске программы

См. определение в файле BotFinderConfigs.py строка 73

6.4.4.6 refresh_samples_on_load

BotFinderConfigs.MainConfig.refresh_samples_on_load

пересоздавать данные выборки из базы данных при каждом запуске программы

См. определение в файле BotFinderConfigs.py строка 70

6.4.4.7 refresh_single_samples_element_if_missing

BotFinderConfigs.MainConfig.refresh_single_samples_element_if_missing

заново загрузить элемент выборки, если он не будет найден в файлах

См. определение в файле BotFinderConfigs.py строка 77

6.4.4.8 report_folder_name

BotFinderConfigs.MainConfig.report_folder_name

название папки для хранения отчета

См. определение в файле BotFinderConfigs.py строка 94

6.4.4.9 samples_folder_name

BotFinderConfigs.MainConfig.samples_folder_name

Название папки для хранения выборок

См. определение в файле BotFinderConfigs.py строка 88

6.4.4.10 samples_player_name_file

BotFinderConfigs.MainConfig.samples_player_name_file

Название файла для хранения выборок игроков

См. определение в файле BotFinderConfigs.py строка 91

6.4.4.11 save_data_base_data

BotFinderConfigs.MainConfig.save_data_base_data

сохранить данные для входа в базу данных

См. определение в файле BotFinderConfigs.py строка 63

6.4.4.12 save_data_base_passwor

BotFinderConfigs.MainConfig.save_data_base_passwor

См. определение в файле BotFinderConfigs.py строка 66

6.4.4.13 save_session_data

BotFinderConfigs.MainConfig.save_session_data

сохранять выборки в файлах

См. определение в файле BotFinderConfigs.py строка 84

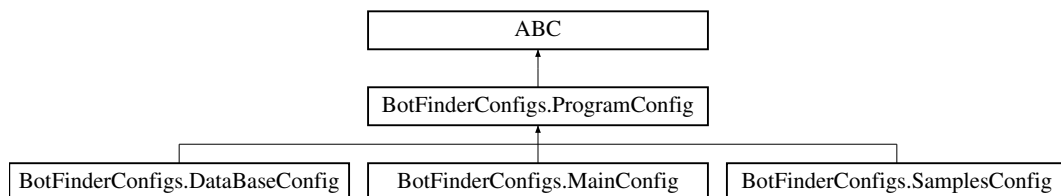
Объявления и описания членов класса находятся в файле:

- [BotFinderConfigs.py](#)

6.5 Класс BotFinderConfigs.ProgramConfig

Абстрактный класс конфигурационных файлов программы

Граф наследования: BotFinderConfigs.ProgramConfig:



Открытые члены

- def [config_file_is_available](#) (self, path=None)
Проверить существует ли файл кофнигурации
- def [default_path](#) (self)
Путь по умолчанию
- def [load_or_create](#) (self, path=None)
Загрузить содержимое файла конфигурации

6.5.1 Подробное описание

Абстрактный класс конфигурационных файлов программы

См. определение в файле BotFinderConfigs.py строка 7

6.5.2 Методы

6.5.2.1 config_file_is_available()

```
def BotFinderConfigs.ProgramConfig.config_file_is_available (
    self,
    path = None )
```

Проверить существует ли файл конфигурации

Аргументы

path	путь к файлу уонфигурации, если не указано, то брется путь по умочанию
------	--

Возвращает

True если файл существует, False если файла нет

См. определение в файле BotFinderConfigs.py строка 16

6.5.2.2 default_path()

```
def BotFinderConfigs.ProgramConfig.default_path (
    self )
```

Путь по умолчанию

Переопределяется в [BotFinderConfigs.SamplesConfig](#), [BotFinderConfigs.DataBaseConfig](#) и [BotFinderConfigs.MainConfig](#)

См. определение в файле BotFinderConfigs.py строка 32

6.5.2.3 load_or_create()

```
def BotFinderConfigs.ProgramConfig.load_or_create (
    self,
    path = None )
```

Загружать содержимое файла конфигурации

Аргументы

path	путь к файлу уонфигурации, если не указано, то брется путь по умочанию
------	--

Возвращает

объект хагруженной конфигурации

См. определение в файле BotFinderConfigs.py строка 43

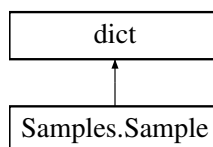
Объявления и описания членов класса находятся в файле:

- [BotFinderConfigs.py](#)

6.6 Класс Samples.Sample

Линейная Выборка данных

Граф наследования:Samples.Sample:



Открытые члены

- `def __init__ (self, name, data_base_request=None)`
- `def median (self)`
получить медиану выборки
- `def quartile (self, number_quartile)`
Получить квартиль линейной выборки
- `def interquartile_range (self)`
получить межквартильное расстояние
- `def import_from_file (self, path)`
Загрузить выборку из файла
- `def export_to_file (self, path)`
записать выборку в файл
- `def try_recover_from_data_base (self, player_id, UserDataBase db)`
- `def linear_ejection (self)`
получить id анамальных элементов выборки
- `def verified_segment (self)`
интервал, в котором находятся достоверные элементы
- `def remissible_segment (self)`
интервал, в которых ожидаемо нохождение некоторого количества элементов
- `def append_player_from_data_base (self, player_id, UserDataBase db)`
добавить элемент выборки из базы данных

Открытые атрибуты

- `name`
- `min`
- `max`
- `data_base_request`

6.6.1 Подробное описание

Линейная Выборка данных

См. определение в файле Samples.py строка 10

6.6.2 Конструктор(ы)

6.6.2.1 `__init__()`

```
def Samples.Sample.__init__(
    self,
    name,
    data_base_request = None )
```

Аргументы

name	имя выборки
data_base_request	запрос базы данных для получения значения выборки по идентификатору пользователя

См. определение в файле Samples.py строка 17

6.6.3 Методы

6.6.3.1 `append_player_from_data_base()`

```
def Samples.Sample.append_player_from_data_base (
    self,
    player_id,
    UserDataBase db )
```

добавить элемент выборки из базы данных

Аргументы

player↔ _id	id элемента выборки
db	база данных

См. определение в файле Samples.py строка 152

6.6.3.2 `export_to_file()`

```
def Samples.Sample.export_to_file (
    self,
    path )
```

записать выборку в файл

Аргументы

path	путь к файлу
------	--------------

См. определение в файле Samples.py строка 82

6.6.3.3 import_from_file()

```
def Samples.Sample.import_from_file (
    self,
    path )
```

Загрузить выборку из файла

Аргументы

path	путь к файлу
------	--------------

См. определение в файле Samples.py строка 69

6.6.3.4 interquartile_range()

```
def Samples.Sample.interquartile_range (
    self )
```

получить межквартильное расстояние

Возвращает

межквартильное расстояние

См. определение в файле Samples.py строка 61

6.6.3.5 linear_ejection()

```
def Samples.Sample.linear_ejection (
    self )
```

получить id аномальных элементов выборки

Возвращает

аномальные элементы

См. определение в файле Samples.py строка 109

6.6.3.6 median()

```
def Samples.Sample.median (  
    self )
```

получить медиану выборки

Возвращает

медиану выборки

См. определение в файле Samples.py строка 34

6.6.3.7 quartile()

```
def Samples.Sample.quartile (  
    self,  
    number_quartile )
```

Получить квартиль линейной выборки

Аргументы

number_quartile	номер квартили
-----------------	----------------

Возвращает

значение квартили

См. определение в файле Samples.py строка 44

6.6.3.8 remissible_segment()

```
def Samples.Sample.remissible_segment (  
    self )
```

интервал, в которых ожидаемо нахождение некоторого количества элементов

Возвращает

два числа, начало и конец интервала

См. определение в файле Samples.py строка 135

6.6.3.9 try_recover_from_data_base()

```
def Samples.Sample.try_recover_from_data_base (
    self,
    player_id,
    UserDataBase db )
```

См. определение в файле Samples.py строка 96

6.6.3.10 verified_segment()

```
def Samples.Sample.verified_segment (
    self )
```

интервал, в котором находятся достоверные элементы

Возвращает

пара значений - начало и конец интервала

См. определение в файле Samples.py строка 119

6.6.4 Данные класса

6.6.4.1 data_base_request

Samples.Sample.data_base_request

См. определение в файле Samples.py строка 25

6.6.4.2 max

Samples.Sample.max

См. определение в файле Samples.py строка 23

6.6.4.3 min

Samples.Sample.min

См. определение в файле Samples.py строка 21

6.6.4.4 name

Samples.Sample.name

См. определение в файле Samples.py строка 19

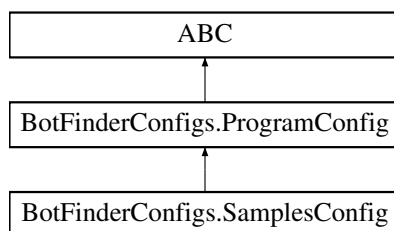
Объявления и описания членов класса находятся в файле:

- [Samples.py](#)

6.7 Класс BotFinderConfigs.SamplesConfig

Конфигурация выборки

Граф наследования: BotFinderConfigs.SamplesConfig:



Открытые члены

- `def __init__(self)`
- `def default_sample(self)`
Заполнить выборки по умолчанию
- `def default_path(self)`
Путь по умолчанию

Открытые атрибуты

- `samples_configs`
выборки
- `user_query_from_data_base`
запрос к базе данных для получения выборки пользователей

Статические открытые данные

- dictionary `sample_model = {'name': None, 'data_base_request': None}`
Конфигурация выборки

6.7.1 Подробное описание

Конфигурация выборки

См. определение в файле BotFinderConfigs.py строка 181

6.7.2 Конструктор(ы)

6.7.2.1 __init__()

```
def BotFinderConfigs.SamplesConfig.__init__(  
    self )
```

См. определение в файле BotFinderConfigs.py строка 186

6.7.3 Методы

6.7.3.1 default_path()

```
def BotFinderConfigs.SamplesConfig.default_path(  
    self )
```

Путь по умолчанию

Переопределяет метод предка [BotFinderConfigs.ProgramConfig](#).

См. определение в файле BotFinderConfigs.py строка 206

6.7.3.2 default_sample()

```
def BotFinderConfigs.SamplesConfig.default_sample(  
    self )
```

Заполнить выборки по умолчанию

См. определение в файле BotFinderConfigs.py строка 198

6.7.4 Данные класса

6.7.4.1 sample_model

```
dictionary BotFinderConfigs.SamplesConfig.sample_model = { 'name': None, 'data_base_request': None } [static]
```

Конфигурация выборки

См. определение в файле BotFinderConfigs.py строка 184

6.7.4.2 samples_configs

BotFinderConfigs.SamplesConfig.samples_configs

выборки

См. определение в файле BotFinderConfigs.py строка 188

6.7.4.3 user_query_from_data_base

BotFinderConfigs.SamplesConfig.user_query_from_data_base

запрос к базе данных для получения выборки пользователей

См. определение в файле BotFinderConfigs.py строка 191

Объявления и описания членов класса находятся в файле:

- [BotFinderConfigs.py](#)

6.8 Класс Samples.ShemaPlayerSamples

класс для комплексного анализа одномерных выборок

Открытые члены

- def `__init__` (self, `points`, `neighbour_count`, `lip`)
- def `append_sample` (self, str name, str query="")
доавить выборку
- def `nearest_distance` (self, point)
вернуть расстояние к ближайшей n-ой точке
- def `distance` (self, first_point, second_pont)
найти нормализованное евклидово расстояние между точками
- def `reachability_distance` (self, first_point, second_pont)
найти достижимое расстояние между двумя точками
- def `point_on_nearest_distance` (self, point)
Получить точки в области досягаемости заданной точки
- def `local_reachability_density` (self, point)
плотность локальной досягаемости заданной точки
- def `local_ouliter_factor` (self, point)
значение локального уровня вброса заданной точки
- def `outliers_point` (self)
получить список аномальных данных методом уровня локального вброса
- def `__getitem__` (self, item)
преобразовать id данных в координаты соответствующей точки
- def `clear_from_incomplete` (self, is_load_from_data_base=True, UserDataBase db=None)
синхронизировать содержание линейных выборок, удалив все элементы, которых нет во всех вы-
борках

Открытые атрибуты

- [points](#)
id точек, в выборках
- [samples](#)
линейные выборки данных
- [neighbour_count](#)
количество соседей в методе локального вброса
- [vp_tree](#)
Дерево точек обзора
- [lip](#)
порог, за которым данные являются анмальными
- [VPtree](#)

6.8.1 Подробное описание

класс для комплексного анализа одномерных выборок

См. определение в файле Samples.py строка 160

6.8.2 Конструктор(ы)

6.8.2.1 `__init__()`

```
def Samples.ShemaPlayerSamples.__init__(
    self,
    points,
    neighbour_count,
    lip )
```

Аргументы

points	id данных в выборках
neighbour_count	количество соседей в методе локального вброса
lip	порог, за которым данные являются анмальными

См. определение в файле Samples.py строка 168

6.8.3 Методы

6.8.3.1 `__getitem__()`

```
def Samples.ShemaPlayerSamples.__getitem__ (
    self,
    item )
```

преобразовать id данных в координаты соответствующей точки

Аргументы

point	идентификатор точки
-------	---------------------

Возвращает

список координат точки

См. определение в файле Samples.py строка 315

6.8.3.2 `append_sample()`

```
def Samples.ShemaPlayerSamples.append_sample (
    self,
    str name,
    str query = "" )
```

доаввить выборку

Аргументы

name	имя выборки
name	запрос выборки

См. определение в файле Samples.py строка 197

6.8.3.3 `clear_from_incomplete()`

```
def Samples.ShemaPlayerSamples.clear_from_incomplete (
    self,
    is_load_from_data_base = True,
    UserDataBase db = None )
```

синхронизировать содержание линейных выборок, удалив все элементы, которых нет во всех выборках

Аргументы

is_load_from_data_base	загружать ли данные из базы данныхесли они не были найдены в файлах
db	база данных для подзагрузки потерянных данных

Возвращает

id удаленных данных

См. определение в файле Samples.py строка 327

6.8.3.4 distance()

```
def Samples.ShemaPlayerSamples.distance (
    self,
    first_point,
    second_pont )
```

найти нормализованное евклидово расстояние между точками

Аргументы

first_point	id первой точки
second_pont	id второй точки

Возвращает

расстояние

См. определение в файле Samples.py строка 223

6.8.3.5 local_outlier_factor()

```
def Samples.ShemaPlayerSamples.local_outlier_factor (
    self,
    point )
```

значение локального уровня вброса заданной точки

Аргументы

point	id заданной точки
-------	-------------------

Возвращает

локальный уровень вброса

См. определение в файле Samples.py строка 280

6.8.3.6 local_reachability_density()

```
def Samples.SchemaPlayerSamples.local_reachability_density (  
    self,  
    point )
```

плотность локальной досягаемости заданной точки

Аргументы

point	id заданной точки
-------	-------------------

Возвращает

значение плотности

См. определение в файле Samples.py строка 260

6.8.3.7 nearest_distance()

```
def Samples.SchemaPlayerSamples.nearest_distance (  
    self,  
    point )
```

вернуть расстояние к ближайшей n-ой точке

Аргументы

point	id набора данных (точки)
-------	--------------------------

Возвращает

список id ближайших данных (точек)

См. определение в файле Samples.py строка 208

6.8.3.8 outliers_point()

```
def Samples.ShemaPlayerSamples.outliers_point (
    self )
```

получить список аномальных данных методом уровня локального вброса

Возвращает

id аномальных данных

См. определение в файле Samples.py строка 291

6.8.3.9 point_on_nearest_distance()

```
def Samples.ShemaPlayerSamples.point_on_nearest_distance (
    self,
    point )
```

Получить точки в области досягаемости заданной точки

Аргументы

point	id заданной точки
-------	-------------------

Возвращает

список id точек в области досягаемости

См. определение в файле Samples.py строка 246

6.8.3.10 reachability_distance()

```
def Samples.ShemaPlayerSamples.reachability_distance (
    self,
    first_point,
    second_pont )
```

найти достижимое расстояние между двумя точками

Аргументы

first_point	id первой точки
second_pont	id второй точки

Возвращает

расстояние

См. определение в файле Samples.py строка 235

6.8.4 Данные класса

6.8.4.1 lip

Samples.SchemaPlayerSamples.lip

порог, за которым данные являются аномальными

См. определение в файле Samples.py строка 182

6.8.4.2 neighbour_count

Samples.SchemaPlayerSamples.neighbour_count

количество соседей в методе локального вброса

См. определение в файле Samples.py строка 176

6.8.4.3 points

Samples.SchemaPlayerSamples.points

id точек, в выборках

См. определение в файле Samples.py строка 170

6.8.4.4 samples

Samples.SchemaPlayerSamples.samples

линейные выборки данных

См. определение в файле Samples.py строка 173

6.8.4.5 vp_tree

Samples.ShemaPlayerSamples.vp_tree

Дерево точек обзора

См. определение в файле Samples.py строка 179

6.8.4.6 VPtree

Samples.ShemaPlayerSamples.VPtree

См. определение в файле Samples.py строка 298

Объявления и описания членов класса находятся в файле:

- [Samples.py](#)

6.9 Класс VPTree.VPTree

Открытые члены

- `def __init__(self, points, distance)`
- `def get_n_nearest_neighbors(self, id_point, count_neighbors)`
Найти n ближайших соседей

Открытые атрибуты

- `left`
левый сосед
- `right`
правый сосед
- `left_min`
минимальное расстояние до потомков в левом поддереве
- `left_max`
максимальное расстояние до потомков в левом поддереве
- `right_min`
минимальное расстояние до потомков в правом поддереве
- `right_max`
максимальное расстояние до потомков в правом поддереве
- `distance`
функция расстояния
- `vantage_point`
идентификатор точки обзора

6.9.1 Подробное описание

См. определение в файле VPTree.py строка 6

6.9.2 Конструктор(ы)

6.9.2.1 `__init__()`

```
def VPTree.VPTree.__init__(
    self,
    points,
    distance )
```

Аргументы

points	идентификаторы точек
distance	функция расстояния

См. определение в файле VPTree.py строка 12

6.9.3 Методы

6.9.3.1 `get_n_nearest_neighbors()`

```
def VPTree.VPTree.get_n_nearest_neighbors (
    self,
    id_point,
    count_neighbors )
```

Найти n ближайших соседей

```
id_point:
count_neighbors:
```

Возвращает

идентификаторы ближайших точек

См. определение в файле VPTree.py строка 81

6.9.4 Данные класса

6.9.4.1 distance

VPTree.VPTree.distance

функция расстояния

См. определение в файле VPTree.py строка 26

6.9.4.2 left

VPTree.VPTree.left

левый сосед

См. определение в файле VPTree.py строка 14

6.9.4.3 left_max

VPTree.VPTree.left_max

максимальное расстояние до потомков в левом поддереве

См. определение в файле VPTree.py строка 20

6.9.4.4 left_min

VPTree.VPTree.left_min

минимальное расстояние до потомков в левом поддереве

См. определение в файле VPTree.py строка 18

6.9.4.5 right

VPTree.VPTree.right

правый сосед

См. определение в файле VPTree.py строка 16

6.9.4.6 right_max

VPTree.VPTree.right_max

максимальное расстояние до потомков в правом поддереве

См. определение в файле VPTree.py строка 24

6.9.4.7 right_min

VPTree.VPTree.right_min

минимальное расстояние до потомков в правом поддереве

См. определение в файле VPTree.py строка 22

6.9.4.8 vantage_point

VPTree.VPTree.vantage_point

идентификатор точки обзора

См. определение в файле VPTree.py строка 28

Объявления и описания членов класса находятся в файле:

- [VPTree.py](#)

Глава 7

Файлы

7.1 Файл BinaryPlot.py

Пространства имен

- [BinaryPlot](#)

Функции

- def [BinaryPlot.plot_local_outlier_factor](#) (names, segments, corrupt_points, clear_points)
построение графика визуально отображающего методов локального выброса и метода тьюки для двух измерений
- def [BinaryPlot.drow_3d_plot](#) (names, segments, corrupt_points, clear_points)
построение графика визуально отображающего методов локального выброса для трех измерений

7.2 Файл BotFinderConfigs.py

Классы

- class [BotFinderConfigs.ProgramConfig](#)
Абстрактный класс конфигурационных файлов программы
- class [BotFinderConfigs.MainConfig](#)
класс основной конфигурации программы
- class [BotFinderConfigs.DataBaseConfig](#)
загрузка файла конфигурации базы данных с диалоговыми опциями
- class [BotFinderConfigs.SamplesConfig](#)
Конфигурация выборки

Пространства имен

- [BotFinderConfigs](#)

Переменные

- `BotFinderConfigs.host`
- `BotFinderConfigs.database`
- `BotFinderConfigs.password`

7.3 Файл main.py

Классы

- class `main.HTMLreport`
класс формирования html отчета
- class `main.BotFinder`
Основной класс программы, реализующий запуск чтения/генерации конфигов программы, функции анализа данных и формирования отчетов

Пространства имен

- `main`

Переменные

- `main.start_status` = `BotFinder.is_first_start()`
- `main.bf` = `BotFinder()`

7.4 Файл Samples.py

Классы

- class `Samples.Sample`
Линейная Выборка данных
- class `Samples.ShemaPlayerSamples`
класс для комплексного анализа одномерных выборок

Пространства имен

- `Samples`

7.5 Файл UserDataBase.py

Пространства имен

- `UserDataBase`

7.6 Файл VPTree.py

Классы

- class `VPTree.VPTree`

Пространства имен

- `VPTree`

Предметный указатель

- `--getitem__`
 - `Samples.SchemaPlayerSamples`, 39
 - `--init__`
 - `BotFinderConfigs.DataBaseConfig`, 18
 - `BotFinderConfigs.MainConfig`, 25
 - `BotFinderConfigs.SamplesConfig`, 37
 - `main.BotFinder`, 14
 - `main.HTMLreport`, 20
 - `Samples.Sample`, 32
 - `Samples.SchemaPlayerSamples`, 39
 - `VPtree.VPtree`, 46
- `append_player_from_data_base`
 - `Samples.Sample`, 32
- `append_sample`
 - `Samples.SchemaPlayerSamples`, 40
- `bf`
 - `main`, 11
- `BinaryPlot`, 9
 - `drow_3d_plot`, 9
 - `plot_local_outlier_factor`, 10
- `BinaryPlot.py`, 49
- `BotFinderConfigs`, 10
 - `database`, 11
 - `host`, 11
 - `password`, 11
- `BotFinderConfigs.DataBaseConfig`, 17
 - `--init__`, 18
 - `database`, 18
 - `default_path`, 18
 - `host`, 18
 - `load_or_create`, 18
 - `password`, 19
 - `user`, 19
- `BotFinderConfigs.MainConfig`, 24
 - `--init__`, 25
 - `count_neighbors`, 25
 - `default_path`, 25
 - `eject_lip`, 25
 - `ignore_error_with_session_element_missing`, 26
 - `parse_samples_from_data_base`, 26
 - `refresh_player_on_load`, 26
 - `refresh_samples_on_load`, 26
 - `refresh_single_samples_element_if_missing`, 26
 - `report_folder_name`, 27
 - `samples_folder_name`, 27
 - `samples_player_name_file`, 27
 - `save_data_base_data`, 27
 - `save_data_base_password`, 27
 - `save_session_data`, 28
- `BotFinderConfigs.ProgramConfig`, 28
 - `config_file_is_available`, 29
 - `default_path`, 30
 - `load_or_create`, 30
- `BotFinderConfigs.py`, 49
- `BotFinderConfigs.SamplesConfig`, 36
 - `--init__`, 37
 - `default_path`, 37
 - `default_sample`, 37
 - `sample_model`, 37
 - `samples_configs`, 37
 - `user_query_from_data_base`, 38
- `clear_from_incomplete`
 - `Samples.SchemaPlayerSamples`, 40
- `config_file_is_available`
 - `BotFinderConfigs.ProgramConfig`, 29
- `count_neighbors`
 - `BotFinderConfigs.MainConfig`, 25
- `data_base`
 - `main.BotFinder`, 16
- `data_base_request`
 - `Samples.Sample`, 35
- `database`
 - `BotFinderConfigs`, 11
 - `BotFinderConfigs.DataBaseConfig`, 18
- `default_path`
 - `BotFinderConfigs.DataBaseConfig`, 18
 - `BotFinderConfigs.MainConfig`, 25
 - `BotFinderConfigs.ProgramConfig`, 30
 - `BotFinderConfigs.SamplesConfig`, 37
- `default_sample`
 - `BotFinderConfigs.SamplesConfig`, 37
- `distance`
 - `Samples.SchemaPlayerSamples`, 41
 - `VPtree.VPtree`, 46
- `drow_3d_plot`
 - `BinaryPlot`, 9
- `eject_lip`
 - `BotFinderConfigs.MainConfig`, 25
- `export_to_file`
 - `Samples.Sample`, 32
- `get_n_nearest_neighbors`

- VPTree.VPTree, 46
- host
 - BotFinderConfigs, 11
 - BotFinderConfigs.DataBaseConfig, 18
- ignore_error_with_session_element_missing
 - BotFinderConfigs.MainConfig, 26
- import_from_file
 - Samples.Sample, 33
- interquartile_range
 - Samples.Sample, 33
- is_first_start
 - main.BotFinder, 14
- left
 - VPTree.VPTree, 47
- left_max
 - VPTree.VPTree, 47
- left_min
 - VPTree.VPTree, 47
- linear_ejection
 - Samples.Sample, 33
- lip
 - Samples.SchemaPlayerSamples, 44
- load_data
 - main.BotFinder, 14
- load_list_of_player
 - main.BotFinder, 14
- load_or_create
 - BotFinderConfigs.DataBaseConfig, 18
 - BotFinderConfigs.ProgramConfig, 30
- load_samples_from_config_file
 - main.BotFinder, 15
- local_ouliter_factor
 - Samples.SchemaPlayerSamples, 41
- local_reachability_density
 - Samples.SchemaPlayerSamples, 42
- main, 11
 - bf, 11
 - start_status, 12
- main.BotFinder, 13
 - __init__, 14
 - data_base, 16
 - is_first_start, 14
 - load_data, 14
 - load_list_of_player, 14
 - load_samples_from_config_file, 15
 - main_config, 16
 - make_report, 15
 - prepare_work_place, 15
 - samples_config, 16
 - shema, 16
 - try_load_data_base, 15
- main.HTMLreport, 19
 - __init__, 20
 - make_report, 20
 - pacage_bloc, 21
 - package_img_bloc, 21
 - package_img_bloc_tuple, 21
 - package_table, 22
 - package_title, 22
 - path_dir, 23
 - prepare_all_image, 23
 - prepare_dir, 23
- main.py, 50
- main_config
 - main.BotFinder, 16
- make_report
 - main.BotFinder, 15
 - main.HTMLreport, 20
- max
 - Samples.Sample, 35
- median
 - Samples.Sample, 33
- min
 - Samples.Sample, 35
- name
 - Samples.Sample, 35
- nearest_distance
 - Samples.SchemaPlayerSamples, 42
- neighbour_count
 - Samples.SchemaPlayerSamples, 44
- outliers_point
 - Samples.SchemaPlayerSamples, 42
- pacage_bloc
 - main.HTMLreport, 21
- package_img_bloc
 - main.HTMLreport, 21
- package_img_bloc_tuple
 - main.HTMLreport, 21
- package_table
 - main.HTMLreport, 22
- package_title
 - main.HTMLreport, 22
- parse_samples_from_data_base
 - BotFinderConfigs.MainConfig, 26
- password
 - BotFinderConfigs, 11
 - BotFinderConfigs.DataBaseConfig, 19
- path_dir
 - main.HTMLreport, 23
- plot_local_ouliter_factor
 - BinaryPlot, 10
- point_on_nearest_distance
 - Samples.SchemaPlayerSamples, 43
- points
 - Samples.SchemaPlayerSamples, 44
- prepare_all_image
 - main.HTMLreport, 23
- prepare_dir
 - main.HTMLreport, 23
- prepare_work_place
 - main.BotFinder, 15

- quartile
 - Samples.Sample, 34
- reachability_distance
 - Samples.SchemaPlayerSamples, 43
- refresh_player_on_load
 - BotFinderConfigs.MainConfig, 26
- refresh_samples_on_load
 - BotFinderConfigs.MainConfig, 26
- refresh_single_samples_element_if_missing
 - BotFinderConfigs.MainConfig, 26
- remissible_segment
 - Samples.Sample, 34
- report_folder_name
 - BotFinderConfigs.MainConfig, 27
- right
 - VPtree.VPtree, 47
- right_max
 - VPtree.VPtree, 47
- right_min
 - VPtree.VPtree, 48
- sample_model
 - BotFinderConfigs.SamplesConfig, 37
- Samples, 12
- samples
 - Samples.SchemaPlayerSamples, 44
- Samples.py, 50
- Samples.Sample, 31
 - __init__, 32
 - append_player_from_data_base, 32
 - data_base_request, 35
 - export_to_file, 32
 - import_from_file, 33
 - interquartile_range, 33
 - linear_ejection, 33
 - max, 35
 - median, 33
 - min, 35
 - name, 35
 - quartile, 34
 - remissible_segment, 34
 - try_recover_from_data_base, 34
 - verified_segment, 35
- Samples.SchemaPlayerSamples, 38
 - __getitem__, 39
 - __init__, 39
 - append_sample, 40
 - clear_from_incomplete, 40
 - distance, 41
 - lip, 44
 - local_outlier_factor, 41
 - local_reachability_density, 42
 - nearest_distance, 42
 - neighbour_count, 44
 - outliers_point, 42
 - point_on_nearest_distance, 43
 - points, 44
 - reachability_distance, 43
 - samples, 44
 - vp_tree, 44
 - VPtree, 45
- samples_config
 - main.BotFinder, 16
- samples_configs
 - BotFinderConfigs.SamplesConfig, 37
- samples_folder_name
 - BotFinderConfigs.MainConfig, 27
- samples_player_name_file
 - BotFinderConfigs.MainConfig, 27
- save_data_base_data
 - BotFinderConfigs.MainConfig, 27
- save_data_base_passwor
 - BotFinderConfigs.MainConfig, 27
- save_session_data
 - BotFinderConfigs.MainConfig, 28
- shema
 - main.BotFinder, 16
- start_status
 - main, 12
- try_load_data_base
 - main.BotFinder, 15
- try_recover_from_data_base
 - Samples.Sample, 34
- user
 - BotFinderConfigs.DataBaseConfig, 19
- user_query_from_data_base
 - BotFinderConfigs.SamplesConfig, 38
- UserDataBase, 12
- UserDataBase.py, 50
- vantage_point
 - VPtree.VPtree, 48
- verified_segment
 - Samples.Sample, 35
- vp_tree
 - Samples.SchemaPlayerSamples, 44
- VPtree, 12
- VPtree
 - Samples.SchemaPlayerSamples, 45
- VPtree.py, 50
- VPtree.VPtree, 45
 - __init__, 46
 - distance, 46
 - get_n_nearest_neighbors, 46
 - left, 47
 - left_max, 47
 - left_min, 47
 - right, 47
 - right_max, 47
 - right_min, 48
 - vantage_point, 48