

Sber Commit Classification Library
0.1.0

Generated by Doxygen 1.9.6

1 Namespace Index	1
1.1 Package List	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 Namespace Documentation	7
4.1 models Namespace Reference	7
4.1.1 Detailed Description	7
4.1.2 Function Documentation	7
4.1.2.1 show_metrics()	7
5 Class Documentation	9
5.1 models.CatboostWithCodeEncoding Class Reference	9
5.1.1 Detailed Description	9
5.1.2 Constructor & Destructor Documentation	9
5.1.2.1 __init__()	10
5.1.3 Member Function Documentation	10
5.1.3.1 code_enc()	10
5.1.3.2 mes_enc()	10
5.1.3.3 predict()	10
5.1.3.4 predict_proba()	11
5.2 Parser.Parser Class Reference	11
5.2.1 Detailed Description	12
5.2.2 Constructor & Destructor Documentation	12
5.2.2.1 __init__()	12
5.2.3 Member Function Documentation	12
5.2.3.1 convert_time()	12
5.2.3.2 count_files()	12
5.2.3.3 count_py_file()	13
5.2.3.4 find_import()	13
5.2.3.5 find_key_word_code()	13
5.2.3.6 find_key_words_message()	13
5.2.3.7 get_all_features_file()	14
5.2.3.8 get_changes()	14
5.2.3.9 get_data()	14
5.2.3.10 get_difference()	14
5.2.3.11 get_merge_commits_file()	15
5.2.3.12 get_merge_commits_web()	15
5.2.3.13 get_merge_requests_file()	15
5.2.3.14 get_path()	15
5.2.3.15 get_project_commits()	16

5.2.3.16	get_project_commits_with_all_features_web()	16
5.2.3.17	get_project_commits_with_basic_features_web()	16
5.2.3.18	get_project_merge_requests_web()	16
5.2.3.19	get_whole_file()	17
5.3	models.SimpleCatboostClassifier Class Reference	17
5.3.1	Detailed Description	17
5.3.2	Constructor & Destructor Documentation	17
5.3.2.1	__init__()	18
5.3.3	Member Function Documentation	18
5.3.3.1	encode_text()	18
5.3.3.2	my_tokenizer()	18
5.3.3.3	predict()	18
5.3.3.4	predict_proba()	19
5.4	models.SimpleCommitDataset Class Reference	19
5.4.1	Detailed Description	19
5.4.2	Constructor & Destructor Documentation	19
5.4.2.1	__init__()	20
5.4.3	Member Function Documentation	20
5.4.3.1	__getitem__()	20
5.4.3.2	__len__()	20
5.5	models.SimpleSberModule Class Reference	20
5.5.1	Detailed Description	21
5.5.2	Constructor & Destructor Documentation	21
5.5.2.1	__init__()	21
5.5.3	Member Function Documentation	22
5.5.3.1	code_enc()	22
5.5.3.2	configure_optimizers()	22
5.5.3.3	fit()	22
5.5.3.4	forward()	23
5.5.3.5	mes_enc()	23
5.5.3.6	predict()	23
5.5.3.7	training_step()	23
5.5.3.8	validation_step()	24
5.6	models.SimpleSberNet Class Reference	24
5.6.1	Detailed Description	24
5.6.2	Constructor & Destructor Documentation	24
5.6.2.1	__init__()	25
5.6.3	Member Function Documentation	25
5.6.3.1	forward()	25
5.7	models.StrongCommitDataset Class Reference	25
5.7.1	Detailed Description	26
5.7.2	Constructor & Destructor Documentation	26
5.7.2.1	__init__()	26

5.7.3 Member Function Documentation	26
5.7.3.1 <code>__getitem__()</code>	26
5.7.3.2 <code>__len__()</code>	27
5.8 <code>models.StrongSberModule</code> Class Reference	27
5.8.1 Detailed Description	28
5.8.2 Constructor & Destructor Documentation	28
5.8.2.1 <code>__init__()</code>	28
5.8.3 Member Function Documentation	28
5.8.3.1 <code>configure_optimizers()</code>	28
5.8.3.2 <code>fit()</code>	29
5.8.3.3 <code>forward()</code>	29
5.8.3.4 <code>training_step()</code>	29
5.8.3.5 <code>validation_step()</code>	30
5.9 <code>models.StrongSberNet</code> Class Reference	30
5.9.1 Detailed Description	30
5.9.2 Constructor & Destructor Documentation	31
5.9.2.1 <code>__init__()</code>	31
5.9.3 Member Function Documentation	31
5.9.3.1 <code>forward()</code>	31
Предметный указатель	33

Глава 1

Namespace Index

1.1 Package List

Here are the packages with brief descriptions (if available):

models	7
------------------	---

Глава 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

models.CatboostWithCodeEncoding	9
pl.LightningModule	
models.SimpleSberModule	20
models.StrongSberModule	27
nn.Module	
models.SimpleSberNet	24
models.StrongSberNet	30
Parser.Parser	11
models.SimpleCatboostClassifier	17
Dataset	
models.SimpleCommitDataset	19
models.StrongCommitDataset	25

Глава 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

models.CatboostWithCodeEncoding	9
Parser.Parser	11
models.SimpleCatboostClassifier	17
models.SimpleCommitDataset	19
models.SimpleSberModule	20
models.SimpleSberNet	24
models.StrongCommitDataset	25
models.StrongSberModule	27
models.StrongSberNet	30

Глава 4

Namespace Documentation

4.1 models Namespace Reference

Classes

- class CatboostWithCodeEncoding
- class SimpleCatboostClassifier
- class SimpleCommitDataset
- class SimpleSberModule
- class SimpleSberNet
- class StrongCommitDataset
- class StrongSberModule
- class StrongSberNet

Functions

- None show_metrics (np.ndarray y_pred, np.ndarray y_true)

4.1.1 Detailed Description

Данный модуль хранит в себе реализации основных моделей для классификации коммитов.

Модели делятся на 2 вида: на основе градиентного бустинга, и на основе нейронных сетей.

4.1.2 Function Documentation

4.1.2.1 show_metrics()

```
None models.show_metrics (  
    np.ndarray y_pred,  
    np.ndarray y_true )
```

Метод для отображения метрик: accuracy, f1_score, roc_auc, precision, recall, f2_score.
:param y_pred: Предсказанные значения.
:param y_true: Реальные значения.

Глава 5

Class Documentation

5.1 models.CatboostWithCodeEncoding Class Reference

Public Member Functions

- `def __init__ (self, bool is_svd=True, str device='cpu')`
- `np.ndarray code_enc (self, pd.DataFrame data)`
- `np.ndarray mes_enc (self, pd.DataFrame data)`
- `np.ndarray predict (self, pd.DataFrame x_data)`
- `np.ndarray predict_proba (self, pd.DataFrame x_data)`

Public Attributes

- `tokenizer`
- `unixcoder`
- `roberta`
- `device`
- `clf`
- `path`
- `decomposition`

5.1.1 Detailed Description

Классификатор использующий числовые данные о коммите и закодированное с помощью нейронных сетей сообщения коммитов и код программы до и после изменения.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 `__init__()`

```
def models.CatboostWithCodeEncoding.__init__(
    self,
    bool is_svd = True,
    str device = 'cpu' )
```

Метод инициализации параметров модели.

:param is_svd: Если True, то применяется метод уменьшения размерности SVD, в противном случае применяется PCA.

:param device: Устройство на котором будут работать нейронные сети для кодирования кода и текста.

5.1.3 Member Function Documentation

5.1.3.1 `code_enc()`

```
np.ndarray models.CatboostWithCodeEncoding.code_enc (
    self,
    pd.DataFrame data )
```

Метод для кодирования кода программы.

:param data: DataFrame с данными о коммите.

:return: Массив закодированных кодов программ до и после.

5.1.3.2 `mes_enc()`

```
np.ndarray models.CatboostWithCodeEncoding.mes_enc (
    self,
    pd.DataFrame data )
```

Метод для кодирования сообщения коммита.

:param data: DataFrame с данными о коммите.

:return: Массив закодированных сообщений коммита.

5.1.3.3 `predict()`

```
np.ndarray models.CatboostWithCodeEncoding.predict (
    self,
    pd.DataFrame x_data )
```

Метод для предсказания значимости коммита.

:param x_data: DataFrame с данными о коммитах.

:return: Массив с метками коммитов(0 - незначимый, 1 - значимый).

5.1.3.4 predict_proba()

```
np.ndarray models.CatboostWithCodeEncoding.predict_proba (
    self,
    pd.DataFrame x_data )
```

Метод для предсказания вероятности значимости коммита.
:param x_data: DataFrame с данными о коммитах.
:return: Массив с вероятностями значимости коммитов.

The documentation for this class was generated from the following file:

- models.py

5.2 Parser.Parser Class Reference

Public Member Functions

- def __init__ (self, token="")
- None get_merge_requests_file (self, str input_file, str output_file)
- None get_merge_commits_file (self, str input_file, str output_file, limit=0)
- None get_all_features_file (self, str input_file, output_file=None)
- def get_changes (self, project_id_, commit_id_)
- def get_data (self, project_id_, commit_id_)
- def get_whole_file (self, project_id, commit_sha, file_names)
- def get_project_commits (self, str project_id)
- def get_project_commits_with_basic_features_web (self, str project_id)
- def get_project_commits_with_all_features_web (self, str project_id)
- def get_project_merge_requests_web (self, str project_id)
- def get_merge_commits_web (self, project_id, iid=-1)

Static Public Member Functions

- def get_difference (changes)
- def count_files (changes)
- def find_import (add)
- def find_key_word_code (code)
- def find_key_words_message (message)
- def get_path (diff)
- def convert_time (date_str)
- def count_py_file (path_files)

Public Attributes

- TOKEN
- HEADERS
- project_id_out_of_func
- t

Static Public Attributes

- `int project_id_out_of_func = 0`
- `pd t = pd.DataFrame()`

5.2.1 Detailed Description

Данный модуль представляет собой парсер коммитов гитлаба.
Реализованы методы, которые взаимодействуют с файлами и с вебом.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 `__init__()`

```
def Parser.Parser.__init__ (
    self,
    token = "" )
```

Функция инициализации парсера, принимает токен для обращения к апи гитлаба
:param token: если токен не указывается, значение по умолчанию - пустая строка,
что соответствует доступу только к открытым проектам

5.2.3 Member Function Documentation

5.2.3.1 `convert_time()`

```
def Parser.Parser.convert_time (
    date_str ) [static]
```

Конвертирует даты к одному формату (гитлаб выдает даты в разном формате).
:param date_str: даты в коммите
:return:

5.2.3.2 `count_files()`

```
def Parser.Parser.count_files (
    changes ) [static]
```

Возвращает кол-во файлов, измененных в коммите
:param changes: diff коммита
:return:

5.2.3.3 count_py_file()

```
def Parser.Parser.count_py_file (
    path_files ) [static]
```

Возвращает кол-во питоновских файлов, измененных в коммите
:param path_files: пути до файлов, измененных в коммите
:return:

5.2.3.4 find_import()

```
def Parser.Parser.find_import (
    add ) [static]
```

Возвращает кол-во добавленных импортов в коммите
:param add: добавленные строки в коммите
:return:

5.2.3.5 find_key_word_code()

```
def Parser.Parser.find_key_word_code (
    code ) [static]
```

Возвращает кол-во ключевых слов (def, class)
:param code: добавленные или удаленные строки в коммите
:return:

5.2.3.6 find_key_words_message()

```
def Parser.Parser.find_key_words_message (
    message ) [static]
```

Возвращает ключевые слов
:param message: commit message
:return:

5.2.3.7 get_all_features_file()

```
None Parser.Parser.get_all_features_file (
    self,
    str input_file,
    output_file = None )
```

Добавляет дополнительные признаки к существующим коммитам
:param input_file: путь до входного файла
:param output_file: путь до выходного файла
:return:

5.2.3.8 get_changes()

```
def Parser.Parser.get_changes (
    self,
    project_id_,
    commit_id_ )
```

Возвращает необходимую информацию (структура словаря: diff, commit_message etc.) полученную по конкретному коммиту
:param project_id_ : id проекта
:param commit_id_ : id коммита
:return:

5.2.3.9 get_data()

```
def Parser.Parser.get_data (
    self,
    project_id_,
    commit_id_ )
```

Возвращает ссылку на коммит на гитлабе, diff коммита
:param project_id_ : id проекта
:param commit_id_ : id коммита
:return:

5.2.3.10 get_difference()

```
def Parser.Parser.get_difference (
    changes ) [static]
```

Возвращает добавленный/удаленный код в коммите на основе diff
:param changes: колонка changes в коммите
:return:

5.2.3.11 get_merge_commits_file()

```
None Parser.Parser.get_merge_commits_file (
    self,
    str input_file,
    str output_file,
    limit = 0 )
```

Функция обрабатывает мердж коммиты и записывает их в файл
:param input_file: путь до входного файла
:param output_file: путь до выходного файла
:param limit: ограничение по коммитам
:return:

5.2.3.12 get_merge_commits_web()

```
def Parser.Parser.get_merge_commits_web (
    self,
    project_id,
    iid = -1 )
```

Возвращает коммиты мердж реквеста с базовыми функциями
если iid равен -1, то возвращает все коммиты мердж реквеста
иначе возвращает коммиты определенного мердж реквеста
:param project_id: id проекта
:param iid: id мердж реквеста
:return:

5.2.3.13 get_merge_requests_file()

```
None Parser.Parser.get_merge_requests_file (
    self,
    str input_file,
    str output_file )
```

Функция достает мердж реквесты из файла
:param input_file: путь до входного файла
:param output_file: путь до выходного файла
:return:

5.2.3.14 get_path()

```
def Parser.Parser.get_path (
    diff ) [static]
```

Возвращает массив путей до файлов, которые были изменены в коммите
:param diff: diff коммита
:return:

5.2.3.15 get_project_commits()

```
def Parser.Parser.get_project_commits (
    self,
    str project_id )
```

Возвращает коммиты проекта
:param project_id: id проекта
:return: список коммитов проекта

5.2.3.16 get_project_commits_with_all_features_web()

```
def Parser.Parser.get_project_commits_with_all_features_web (
    self,
    str project_id )
```

Возвращает коммиты проекта со всеми признаками
:param project_id: id проекта
:return: датафрейм со всеми признаками

5.2.3.17 get_project_commits_with_basic_features_web()

```
def Parser.Parser.get_project_commits_with_basic_features_web (
    self,
    str project_id )
```

Возвращает коммиты проекта с базовыми признаками в виде pandas DataFrame.
:param project_id: id проекта
:return: датафрейм с базовыми признаками

5.2.3.18 get_project_merge_requests_web()

```
def Parser.Parser.get_project_merge_requests_web (
    self,
    str project_id )
```

Вернуть мердж реквесты проекта
:param project_id: id проекта
:return: список мердж реквестов

5.2.3.19 get_whole_file()

```
def Parser.Parser.get_whole_file (
    self,
    project_id,
    commit_sha,
    file_names )
```

Возвращает изначальный код файлов в виде одной строки, новый код файлов в виде одной строки
массив: путь до файла - его код (до и после коммита), проверка на совпадение кода "до" и "после" коммита
:param project_id: id проекта
:param commit_sha: sha коммита
:param file_names: пути до файлов измененных в этом коммите
:return:

The documentation for this class was generated from the following file:

- Parser.py

5.3 models.SimpleCatboostClassifier Class Reference

Public Member Functions

- def `__init__` (self, is_bow=False)
- List[str] `my_tokenizer` (self, str text)
- np.ndarray `encode_text` (self, pd.DataFrame data)
- np.ndarray `predict` (self, pd.DataFrame x_data)
- np.ndarray `predict_proba` (self, pd.DataFrame x_data)

Public Attributes

- tokenizer
- clf
- path
- is_bow

5.3.1 Detailed Description

Самый простой классификатор использует только числовые данные о коммите и закодированное с помощью TF-IDF/Bag of Words сообщение коммита.

5.3.2 Constructor & Destructor Documentation

5.3.2.1 `__init__()`

```
def models.SimpleCatboostClassifier.__init__ (
    self,
    is_bow = False )
```

Функция инициализирующая параметры модели.

:param is_bow: Если True, то показывает, что для векторизации токенов будет использоваться BoW, а если False, то TF-IDF.

5.3.3 Member Function Documentation

5.3.3.1 `encode_text()`

```
np.ndarray models.SimpleCatboostClassifier.encode_text (
    self,
    pd.DataFrame data )
```

Метод для кодирования токенов в текст.

:param data: DataFrame с данными о коммитах.

:return: Массив с векторными представлениями сообщений коммита.

5.3.3.2 `my_tokenizer()`

```
List[str] models.SimpleCatboostClassifier.my_tokenizer (
    self,
    str text )
```

Метод для разбиения текста на токены.

:param text: Текст для токенизации.

:return: Список токенов.

5.3.3.3 `predict()`

```
np.ndarray models.SimpleCatboostClassifier.predict (
    self,
    pd.DataFrame x_data )
```

Метод для предсказания значимости коммита.

:param x_data: DataFrame с данными о коммитах.

:return: Массив с метками коммитов(0 - незначимый, 1 - значимый).

5.3.3.4 predict_proba()

```
np.ndarray models.SimpleCatboostClassifier.predict_proba (
    self,
    pd.DataFrame x_data )
```

Метод для предсказания вероятности значимости коммита.

:param x_data: DataFrame с данными о коммитах.

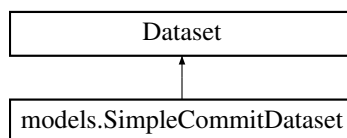
:return: Массив с вероятностями значимости коммитов.

The documentation for this class was generated from the following file:

- models.py

5.4 models.SimpleCommitDataset Class Reference

Inheritance diagram for models.SimpleCommitDataset:



Public Member Functions

- def `__init__` (self, np.ndarray x, np.ndarray y, str device='cpu')
- Tuple[torch.Tensor, torch.Tensor] `__getitem__` (self, int idx)
- int `__len__` (self)

Public Attributes

- x
- y
- device

5.4.1 Detailed Description

Класс датасета для хранения и выдачи информации для модели SimpleSberNet.

5.4.2 Constructor & Destructor Documentation

5.4.2.1 `__init__()`

```
def models.SimpleCommitDataset.__init__(
    self,
    np.ndarray x,
    np.ndarray y,
    str device = 'cpu' )
```

Метод инициализации параметров датасета.

:param x: Признаки коммита.

:param y: Целевая переменная.

:param device: Устройство на котором должны находиться тензоры.

5.4.3 Member Function Documentation

5.4.3.1 `__getitem__()`

```
Tuple[torch.Tensor, torch.Tensor] models.SimpleCommitDataset.__getitem__(
    self,
    int idx )
```

Метод выдающий данные о признаках и целевой метке коммита под индексом `idx`.

:param idx: Индекс коммита.

:return: Тензор признаков, и тензор целевой метки.

5.4.3.2 `__len__()`

```
int models.SimpleCommitDataset.__len__(
    self )
```

Метод для возвращения длины датасета.

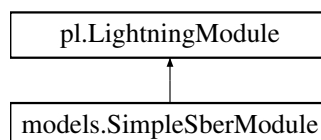
:return: Количество коммитов в датасете.

The documentation for this class was generated from the following file:

- `models.py`

5.5 `models.SimpleSberModule` Class Reference

Inheritance diagram for `models.SimpleSberModule`:



Public Member Functions

- `def __init__(self, nn.Module loss_func=nn.BCEWithLogitsLoss(pos_weight=torch.tensor([3])), lr_func=None, bool is_pretrained=False, str device='cpu', *args, **kwargs)`
- `torch.Tensor forward (self, torch.Tensor x)`
- `np.ndarray predict (self, pd.DataFrame x_data)`
- `Tuple[DataLoader, DataLoader] fit (self, pd.DataFrame train_x, np.ndarray train_y, pd.DataFrame val_x, np.ndarray val_y, torch.optim.Optimizer optimizer, str accelerator='cpu', int max_epochs=20, int batch_size=32)`
- `np.ndarray code_enc (self, pd.DataFrame data)`
- `np.ndarray mes_enc (self, pd.DataFrame data)`
- `Union[torch.optim.Optimizer, Tuple[List[torch.optim.Optimizer], List[torch.optim.lr_scheduler.LambdaLR]]] configure_optimizers (self)`
- `torch.Tensor training_step (self, tuple train_batch, int batch_idx)`
- `def validation_step (self, tuple val_batch, int batch_idx)`

Public Attributes

- `dev`
- `tokenizer`
- `unixcoder`
- `roberta`
- `lr_func`
- `loss_func`
- `optim`
- `path`
- `accuracy`
- `f1`
- `roc_auc`
- `clf`

5.5.1 Detailed Description

Класс для обучения модели SimpleSberNet.

5.5.2 Constructor & Destructor Documentation

5.5.2.1 `__init__()`

```
def models.SimpleSberModule.__init__(
    self,
    nn.Module loss_func = nn.BCEWithLogitsLoss(pos_weight=torch.tensor([3])),
    lr_func = None,
    bool is_pretrained = False,
    str device = 'cpu',
    * args,
    ** kwargs )
```

Метод инициализации параметров объекта класса.

:param loss_func: Функция потерь.

:param lr_func: Функция уменьшения шага обучения.

:param is_pretrained: Если True, то можно будет применять предобученную модель, в противном случае модель будет новая.

:param device: Устройство на котором будет работать нейронная сеть.

:param args: -

:param kwargs: -

5.5.3 Member Function Documentation

5.5.3.1 code_enc()

```
np.ndarray models.SimpleSberModule.code_enc (
    self,
    pd.DataFrame data )
```

Метод для кодирования кода программы.
:param data: DataFrame с данными о коммите.
:return: Массив закодированных кодов программ до и после.

5.5.3.2 configure_optimizers()

```
Union[torch.optim.Optimizer, Tuple[List[torch.optim.Optimizer], List[torch.optim.lr_scheduler.LambdaLR]]] models.SimpleSberModule.configure_optimizers (
    self )
```

Метод для возврата оптимизатора.
:return: Оптимизатор.

5.5.3.3 fit()

```
Tuple[DataLoader, DataLoader] models.SimpleSberModule.fit (
    self,
    pd.DataFrame train_x,
    np.ndarray train_y,
    pd.DataFrame val_x,
    np.ndarray val_y,
    torch.optim.Optimizer optimizer,
    str accelerator = 'cpu',
    int max_epochs = 20,
    int batch_size = 32 )
```

Метод для обучения модели.
:param train_x: Тренировочные признаки коммитов.
:param train_y: Тренировочные целевые метки коммитов.
:param val_x: Валидационные признаки коммитов.
:param val_y: Валидационные целевые метки коммитов
:param optimizer: Оптимизатор градиентного спуска.
:param accelerator: Устройство, на котором будет происходить обучение.
:param max_epochs: Количество эпох обучения.
:param batch_size: Размер batch.
:return: Тренировочный и валидационный dataloader.

5.5.3.4 forward()

```
torch.Tensor models.SimpleSberModule.forward (
    self,
    torch.Tensor x )
```

Метод для предсказания.
:param x: Тензор признаков коммитов.
:return: Предсказание вероятности значимости коммита.

5.5.3.5 mes_enc()

```
np.ndarray models.SimpleSberModule.mes_enc (
    self,
    pd.DataFrame data )
```

Метод для кодирования сообщения коммита.
:param data: DataFrame с данными о коммите.
:return: Массив закодированных сообщений коммита.

5.5.3.6 predict()

```
np.ndarray models.SimpleSberModule.predict (
    self,
    pd.DataFrame x_data )
```

Метод для предсказания значимости коммита.
:param x_data: DataFrame с данными о коммитах.
:return: Массив с вероятностью значимости коммита.

5.5.3.7 training_step()

```
torch.Tensor models.SimpleSberModule.training_step (
    self,
    tuple train_batch,
    int batch_idx )
```

Метод одной итерации шага обучения.
:param train_batch: Batch.
:param batch_idx: Индекс batch.
:return: Ошибку

5.5.3.8 validation_step()

```
def models.SimpleSberModule.validation_step (
    self,
    tuple val_batch,
    int batch_idx )
```

Метод одной итерации шага валидации.

:param val_batch: Batch.

:param batch_idx: Индекс batch.

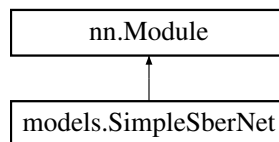
:return: Ошибку

The documentation for this class was generated from the following file:

- models.py

5.6 models.SimpleSberNet Class Reference

Inheritance diagram for models.SimpleSberNet:



Public Member Functions

- def __init__ (self)
- torch.Tensor forward (self, torch.Tensor x)

Public Attributes

- bn0
- fc1
- bn1
- relu
- fc2
- bn2
- classifier

5.6.1 Detailed Description

Класс нейронной сети.

5.6.2 Constructor & Destructor Documentation

5.6.2.1 `__init__()`

```
def models.SimpleSberNet.__init__(
    self )
```

Класс для инициализации основных слоёв сети.

5.6.3 Member Function Documentation

5.6.3.1 `forward()`

```
torch.Tensor models.SimpleSberNet.forward (
    self,
    torch.Tensor x )
```

Метод для предсказания.

:param x: Тензор признаков коммитов.

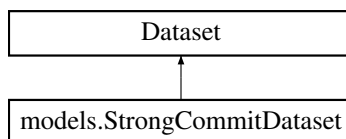
:return: Предсказание вероятности значимости коммита.

The documentation for this class was generated from the following file:

- models.py

5.7 models.StrongCommitDataset Class Reference

Inheritance diagram for models.StrongCommitDataset:



Public Member Functions

- `def __init__(self, pd.DataFrame x, np.ndarray y, transformers.models.roberta.tokenization_ ↵
roberta.RobertaTokenizer tokenizer, nn.Module mess_model, nn.Module code_model, int max_ ↵
len=64)`
- `def __getitem__(self, int idx)`
- `def __len__(self)`

Public Attributes

- x
- y
- tokenizer
- mess_model
- code_model
- max_len

5.7.1 Detailed Description

Класс датасета для хранения и выдачи информации о коммите для модели StrongSberNet.

5.7.2 Constructor & Destructor Documentation

5.7.2.1 __init__()

```
def models.StrongCommitDataset.__init__(
    self,
    pd.DataFrame x,
    np.ndarray y,
    transformers.models.roberta.tokenization_roberta.RobertaTokenizer tokenizer,
    nn.Module mess_model,
    nn.Module code_model,
    int max_len = 64 )
```

Метод инициализации датасета.

:param x: Признаки коммитов.

:param y: Целевые метки коммитов.

:param tokenizer: Токенизатор для сообщения коммита.

:param mess_model: Модель для кодирования сообщения коммита.

:param code_model: Модель для кодирования кода до и после изменения.

:param max_len: Максимальная длина эмбединга скрытого представления Encoder слоя модели.

5.7.3 Member Function Documentation

5.7.3.1 __getitem__()

```
def models.StrongCommitDataset.__getitem__(
    self,
    int idx )
```

Метод выдающий данные о признаках и целевой метке коммита под индексом idx.

:param idx: Индекс коммита.

:return: Тензор признаков, и тензор целевой метки.

5.7.3.2 `__len__()`

```
def models.StrongCommitDataset.__len__(
    self )
```

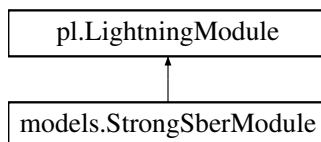
Метод для возвращения длины датасета.
:return: Количество коммитов в датасете.

The documentation for this class was generated from the following file:

- models.py

5.8 models.StrongSberModule Class Reference

Inheritance diagram for models.StrongSberModule:



Public Member Functions

- `def __init__(self, nn.Module loss_func=nn.BCEWithLogitsLoss(pos_weight=torch.tensor([2])), lr_func=None, bool is_full=False, str device='cpu', *args, **kwargs)`
- `torch.Tensor forward(self, Dict[str, torch.Tensor] x)`
- `Tuple[DataLoader, DataLoader] fit(self, pd.DataFrame train_x, np.ndarray train_y, pd.DataFrame val_x, np.ndarray val_y, torch.optim.Optimizer optimizer, str accelerator='cpu', int max_epochs=20, int batch_size=4, int max_len=64)`
- `Union[torch.optim.Optimizer, Tuple[List[torch.optim.Optimizer], List[torch.optim.lr_scheduler.LambdaLR]]] configure_optimizers(self)`
- `def training_step(self, Dict[str, torch.Tensor] train_batch, int batch_idx)`
- `def validation_step(self, Dict[str, torch.Tensor] val_batch, int batch_idx)`

Public Attributes

- `dev`
- `optim`
- `lr_func`
- `loss_func`
- `path`
- `tokenizer`
- `accuracy`
- `f1`
- `roc_auc`
- `clf`

5.8.1 Detailed Description

Класс для обучения модели StrongSberNet.

5.8.2 Constructor & Destructor Documentation

5.8.2.1 `__init__()`

```
def models.StrongSberModule.__init__(
    self,
    nn.Module loss_func = nn.BCEWithLogitsLoss(pos_weight=torch.tensor([2])),
    lr_func = None,
    bool is_full = False,
    str device = 'cpu',
    * args,
    ** kwargs )
```

Класс инициализации.

:param loss_func: Функция потерь.

:param lr_func: Функция уменьшения шага обучения.

:param is_full: Если True, то будут обучаться все модели, которые участвуют в кодировании кода и сообщений коммита, в противном случае, только часть данных моделей.

:param device: Устройство, на котором будет работать модель.

:param args: -

:param kwargs: -

5.8.3 Member Function Documentation

5.8.3.1 `configure_optimizers()`

Union[torch.optim.Optimizer, Tuple[List[torch.optim.Optimizer], List[torch.optim.lr_scheduler.LambdaLR]]] models.↔

```
StrongSberModule.configure_optimizers (
    self )
```

Метод для возврата оптимизатора.

:return: Оптимизатор.

5.8.3.2 fit()

```

Tuple[DataLoader, DataLoader] models.StrongSberModule.fit (
    self,
    pd.DataFrame train_x,
    np.ndarray train_y,
    pd.DataFrame val_x,
    np.ndarray val_y,
    torch.optim.Optimizer optimizer,
    str accelerator = 'cpu',
    int max_epochs = 20,
    int batch_size = 4,
    int max_len = 64 )

```

Метод для обучения модели StrongSberNet.

```

:param train_x: Тренировочные признаки коммитов.
:param train_y: Тренировочные целевые метки коммитов.
:param val_x: Валидационные признаки коммитов.
:param val_y: Валидационные целевые метки коммитов
:param optimizer: Оптимизатор градиентного спуска.
:param accelerator: Устройство, на котором будет происходить обучение.
:param max_epochs: Количество эпох обучения.
:param batch_size: Размер batch.
:param max_len: Максимальная длина массива токенов.
:return: Тренировочный и валидационный dataloader.

```

5.8.3.3 forward()

```

torch.Tensor models.StrongSberModule.forward (
    self,
    Dict[str, torch.Tensor] x )

```

Метод для предсказания.

```

:param x: Словарь признаков коммитов.
:return: Предсказание вероятности значимости коммита.

```

5.8.3.4 training_step()

```

def models.StrongSberModule.training_step (
    self,
    Dict[str, torch.Tensor] train_batch,
    int batch_idx )

```

Метод одной итерации шага обучения.

```

:param train_batch: Batch.
:param batch_idx: Индекс batch.
:return: Ошибку

```

5.8.3.5 validation_step()

```
def models.StrongSberModule.validation_step (
    self,
    Dict[str, torch.Tensor] val_batch,
    int batch_idx )
```

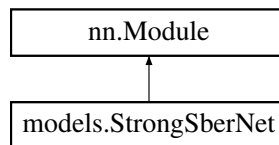
Метод одной итерации шага валидации.
:param val_batch: Batch.
:param batch_idx: Индекс batch.
:return: Ошибку

The documentation for this class was generated from the following file:

- models.py

5.9 models.StrongSberNet Class Reference

Inheritance diagram for models.StrongSberNet:



Public Member Functions

- def `__init__` (self, bool is_full)
- torch.Tensor `forward` (self, Dict[str, torch.Tensor] x)

Public Attributes

- bn1
- relu
- fc2
- bn2
- classifier
- code_model
- mess_model
- bn0
- fc1

5.9.1 Detailed Description

Класс нейронной сети.

5.9.2 Constructor & Destructor Documentation

5.9.2.1 `__init__()`

```
def models.StrongSberNet.__init__(  
    self,  
    bool is_full )
```

Класс нейронной сети

:param is_full: Если True, то будут обучаться все модели, которые участвуют в кодировании кода и сообщений коммита, в противном случае, только часть данных моделей

5.9.3 Member Function Documentation

5.9.3.1 `forward()`

```
torch.Tensor models.StrongSberNet.forward (  
    self,  
    Dict[str, torch.Tensor] x )
```

Метод для предсказания.

:param x: Словарь признаков коммитов.

:return: Предсказание вероятности значимости коммита.

The documentation for this class was generated from the following file:

- models.py

Предметный указатель

- `--getitem__`
 - `models.SimpleCommitDataset`, 20
 - `models.StrongCommitDataset`, 26
- `--init__`
 - `models.CatboostWithCodeEncoding`, 9
 - `models.SimpleCatboostClassifier`, 17
 - `models.SimpleCommitDataset`, 19
 - `models.SimpleSberModule`, 21
 - `models.SimpleSberNet`, 24
 - `models.StrongCommitDataset`, 26
 - `models.StrongSberModule`, 28
 - `models.StrongSberNet`, 31
 - `Parser.Parser`, 12
- `--len__`
 - `models.SimpleCommitDataset`, 20
 - `models.StrongCommitDataset`, 26
- `code_enc`
 - `models.CatboostWithCodeEncoding`, 10
 - `models.SimpleSberModule`, 22
- `configure_optimizers`
 - `models.SimpleSberModule`, 22
 - `models.StrongSberModule`, 28
- `convert_time`
 - `Parser.Parser`, 12
- `count_files`
 - `Parser.Parser`, 12
- `count_py_file`
 - `Parser.Parser`, 12
- `encode_text`
 - `models.SimpleCatboostClassifier`, 18
- `find_import`
 - `Parser.Parser`, 13
- `find_key_word_code`
 - `Parser.Parser`, 13
- `find_key_words_message`
 - `Parser.Parser`, 13
- `fit`
 - `models.SimpleSberModule`, 22
 - `models.StrongSberModule`, 28
- `forward`
 - `models.SimpleSberModule`, 22
 - `models.SimpleSberNet`, 25
 - `models.StrongSberModule`, 29
 - `models.StrongSberNet`, 31
- `get_all_features_file`
 - `Parser.Parser`, 13
- `get_changes`
 - `Parser.Parser`, 14
- `get_data`
 - `Parser.Parser`, 14
- `get_difference`
 - `Parser.Parser`, 14
- `get_merge_commits_file`
 - `Parser.Parser`, 14
- `get_merge_commits_web`
 - `Parser.Parser`, 15
- `get_merge_requests_file`
 - `Parser.Parser`, 15
- `get_path`
 - `Parser.Parser`, 15
- `get_project_commits`
 - `Parser.Parser`, 15
- `get_project_commits_with_all_features_web`
 - `Parser.Parser`, 16
- `get_project_commits_with_basic_features_web`
 - `Parser.Parser`, 16
- `get_project_merge_requests_web`
 - `Parser.Parser`, 16
- `get_whole_file`
 - `Parser.Parser`, 16
- `mes_enc`
 - `models.CatboostWithCodeEncoding`, 10
 - `models.SimpleSberModule`, 23
- `models`, 7
 - `show_metrics`, 7
- `models.CatboostWithCodeEncoding`, 9
 - `--init__`, 9
 - `code_enc`, 10
 - `mes_enc`, 10
 - `predict`, 10
 - `predict_proba`, 10
- `models.SimpleCatboostClassifier`, 17
 - `--init__`, 17
 - `encode_text`, 18
 - `my_tokenizer`, 18
 - `predict`, 18
 - `predict_proba`, 18
- `models.SimpleCommitDataset`, 19
 - `--getitem__`, 20
 - `--init__`, 19
 - `--len__`, 20
- `models.SimpleSberModule`, 20
 - `--init__`, 21
 - `code_enc`, 22

- configure_optimizers, 22
 - fit, 22
 - forward, 22
 - mes_enc, 23
 - predict, 23
 - training_step, 23
 - validation_step, 23
- models.SimpleSberNet, 24
 - __init __, 24
 - forward, 25
- models.StrongCommitDataset, 25
 - __getitem __, 26
 - __init __, 26
 - __len __, 26
- models.StrongSberModule, 27
 - __init __, 28
 - configure_optimizers, 28
 - fit, 28
 - forward, 29
 - training_step, 29
 - validation_step, 29
- models.StrongSberNet, 30
 - __init __, 31
 - forward, 31
- my_tokenizer
 - models.SimpleCatboostClassifier, 18
- Parser.Parser, 11
 - __init __, 12
 - convert_time, 12
 - count_files, 12
 - count_py_file, 12
 - find_import, 13
 - find_key_word_code, 13
 - find_key_words_message, 13
 - get_all_features_file, 13
 - get_changes, 14
 - get_data, 14
 - get_difference, 14
 - get_merge_commits_file, 14
 - get_merge_commits_web, 15
 - get_merge_requests_file, 15
 - get_path, 15
 - get_project_commits, 15
 - get_project_commits_with_all_features_web, 16
 - get_project_commits_with_basic_features_web, 16
 - get_project_merge_requests_web, 16
 - get_whole_file, 16
- predict
 - models.CatboostWithCodeEncoding, 10
 - models.SimpleCatboostClassifier, 18
 - models.SimpleSberModule, 23
- predict_proba
 - models.CatboostWithCodeEncoding, 10
 - models.SimpleCatboostClassifier, 18
- show_metrics
 - models, 7
 - training_step
 - models.SimpleSberModule, 23
 - models.StrongSberModule, 29
 - validation_step
 - models.SimpleSberModule, 23
 - models.StrongSberModule, 29