Natural Language Processing Course

Autumn 2023 Stream 5

15 October 2023

mllibs

NLP assisted ML

Цель Проекта

mllibs проектная работа

Создания инструмента который позволит пользователю (**без знании программирования**) выполнять проекты **машинного обучения** используя **текстовые запросы**

mllibs
NLP assisted ML

Цель Проекта

mllibs проектная работа

Создания инструмента который позволит пользователю (**без знании программирования**) выполнять проекты **машинного обучения** используя **текстовые запросы**

Интерпретатор (и **исполнитель**) 1

Минимальное или без знании написания кода

Генераторы 2

Есть знание написания кода

mllibs

NLP assisted ML

Цель Проекта

mllibs проектная работа

Запрос

" ... "

Интерпретатор
(и **исполнитель**)

1

Выполнение запроса

```python
import pandas as pd

def read_csv(path:str):
    pd.read_csv(path)
```

Запрос

" ... "

which pandas function can I use to drop missing data

Генераторы

2

You can use the dropna() function in pandas to drop missing data.

```python
import pandas as pd

def drop_na(data):
    data.dropna()
```

```python
import seaborn as sns

def plot_scatter(data,x,y):
    sns.scatterplot(data,x,y)
```

mllibs

NLP assisted ML

Этапы ML проекта

mllibs проектная работа

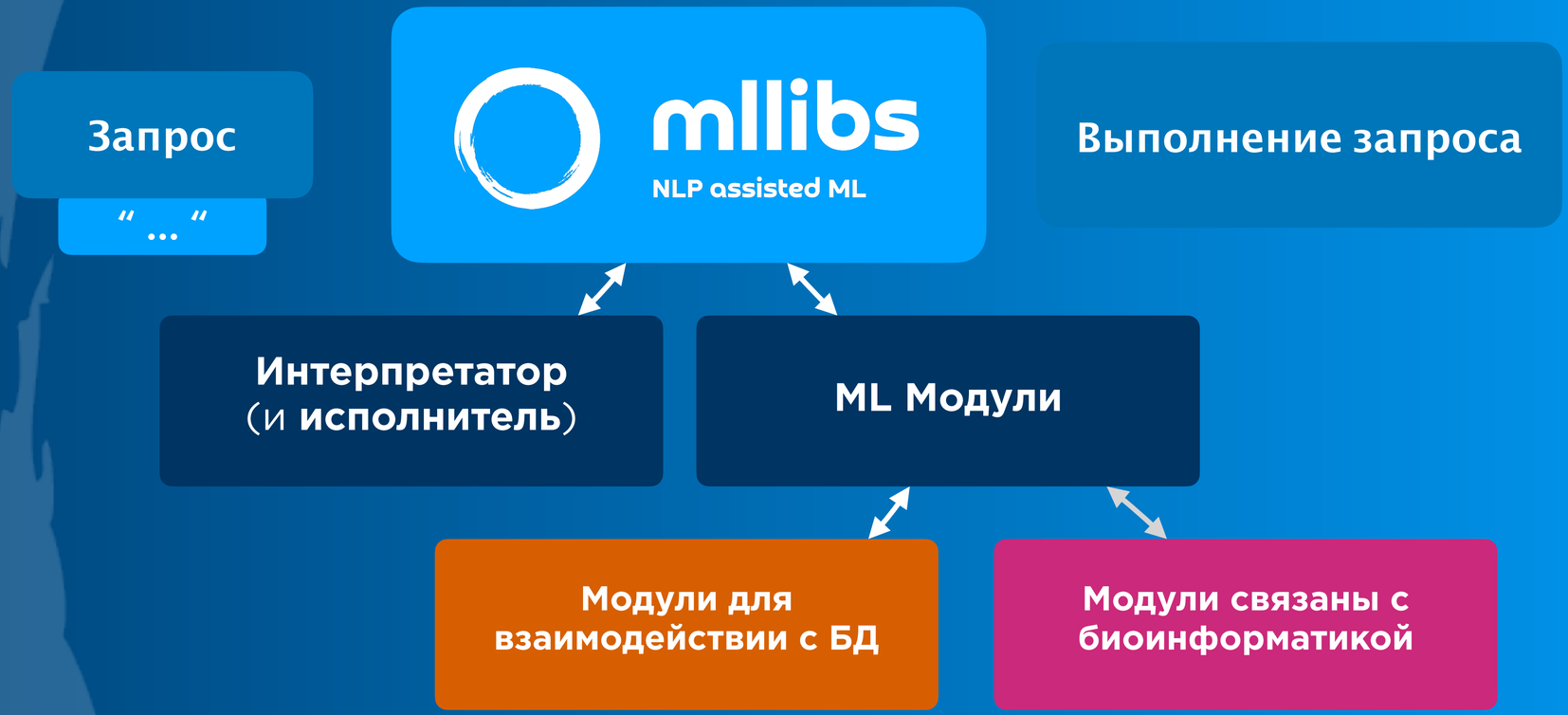
1. Выбор модели: на этом этапе выбирается модель машинного обучения, которая будет использоваться для решения задачи
2. Сбор и подготовка данных: в этом этапе происходит сбор данных, их анализ и очистка от ошибок и выбросов
3. Обучение модели: в этом этапе происходит обучение модели на тренировочных данных
4. Оценка модели: после обучения модели необходимо оценить ее качество на тестовых данных
5. Тюнинг гиперпараметров: на этом этапе происходит подбор оптимальных значений гиперпараметров модели
6. Развертывание модели: после успешного обучения и оценки модели ее можно развернуть в продакшн

mllibs

NLP assisted ML

Зачем нужен такое инструмент

mllibs проектная работа



mllibs
NLP assisted ML

Обзор Интерпретатора

mllibs
NLP assisted ML

Методы Активизации Кода

mllibs проектная работа

Функция

```python
def fib_list(n):
    result = []
    a,b = 0,1
    while a<n:
        result.append(a)
        a,b = b, a + b
    return result

fib_list(5)
```

Класс

```python
class fib_list:

    def __init__(self,n):
        self.n = n

    def get_list(self):
        result = []
        a,b = 0,1
        while a<self.n:
            result.append(a)
            a,b = b, a + b
        return result

fib = fib_list(5)
fib.get_list()
```

Текст

```
input = 'calculate the fibonacci
         sequence for the value
of 5'

nlp_interpreter(input)
```

mllibs

NLP assisted ML

Как это можно реализовать

mllibs проектная работа

mllibs
NLP assisted ML

Как это можно реализовать

mllibs проектная работа

mllibs
NLP assisted ML

Функции активации

mllibs проектная работа

Сбор и подготовка данных: в этом этапе происходит сбор данных, их анализ и очистка от ошибок и выбросов

Загрузка данных

```python
import pandas as pd

def read_csv(path:str):
    pd.read_csv(path)
```

EDA

```python
import seaborn as sns

def plot_scatter(data,x,y):
    sns.scatterplot(data,x,y)
```

Пропуски

```python
import pandas as pd

def drop_na(data):
    data.dropna()
```

Выбросы

```python
import pandas as pd
import numpy as np

def drop_na(data):

    mean = np.mean(data)
    std = np.std(data)

    threshold = 3
    outlier = []
    for i in data:
        z = (i-mean)/std
        if z > threshold:
            outlier.append(i)
    print('outlier in dataset is', outlier)
```

mllibs

NLP assisted ML

Функции активации и метки

mllibs проектная работа

Сбор и подготовка данных: в этом этапе происходит сбор данных, их анализ и очистка от ошибок и выбросов

Загрузка данных | EDA | Пропуски | Выбросы

```python
import pandas as pd

def read_csv(path:str):
    pd.read_csv(path)
```

load_csv

```python
import seaborn as sns

def plot_scatter(data,x,y):
    sns.scatterplot(data,x,y)
```

eda_scatter

```python
import pandas as pd

def drop_na(data):
    data.dropna()
```

pp_drop

```python
import pandas as pd
import numpy as np

def drop_na(data):

    mean = np.mean(data)
    std = np.std(data)

    threshold = 3
    outlier = []
    for i in data:
        z = (i-mean)/std
        if z > threshold:
            outlier.append(i)
    print('outlier in dataset is', outlier)
```

pp_outlier

mllibs

NLP assisted ML

NLP подходы для вызова функции

mllibs проектная работа

NLP задачи которые могут вызвать функцию

Классификация

User пишет запрос в текстовом формате и мы предсказываем метку выполняя классификацию текста

Генерации Текста

User пишет запрос в текстовом формате и мы используем это как контекст, прогоняем через (Transformer Decoder/Seq-Seq) модель и получаем генерированный текст содержащий метку (или метки)

mllibs
NLP assisted ML

NLP подходы для вызова функции

mllibs проектная работа

NLP задачи которые могут вызвать функцию

Обучение

Классификация

User пишет запрос в текстовом формате и мы предсказываем метку выполняя классификацию текста

“Загрузка CSV”
“Загрузите CSV”
“Загрузите CSV используя Pandas”

load_csv

“Загрузка JSON”
“Загрузите JSON”

load_json

Предобработка текста

Токенизация + BOW
Токенизация + Word2Vec
...

RandomForestClassifier()

mllibs
NLP assisted ML

NLP подходы для вызова функции

mllibs проектная работа

NLP задачи которые могут вызвать функцию

Inference

Классификация

User пишет запрос в текстовом формате и мы предсказываем метку выполняя классификацию текста

“Загрузите мне CSV”

Предобработка текста

Токенизация + BOW
Токенизация + Word2Vec
...

RandomForestClassifier()

load_csv

mllibs
NLP assisted ML

NLP подходы для вызова функции

mllibs проектная работа

NLP задачи которые могут вызвать функцию

Генерации Текста

User пишет запрос в текстовом формате и мы используем это как контекст, прогоняем через (Transformer Decoder/Seq-Seq) модель и получаем генерированный текст содержащий метку (или метки)

```python
from transformers import GPT2LMHeadModel, GPT2Tokenizer

model_name_or_path = "sberbank-ai/rugpt3large_based_on_gpt2"
tokenizer = GPT2Tokenizer.from_pretrained(model_name_or_path)
model = GPT2LMHeadModel.from_pretrained(model_name_or_path).cuda()

#вводный текст
text = "Александр Сергеевич Пушкин родился в "

input_ids = tokenizer.encode(text, return_tensors="pt").cuda()
out = model.generate(input_ids.cuda())
generated_text = list(map(tokenizer.decode, out))[0]

# сгенерированный текст
print(generated_text)
# Александр Сергеевич Пушкин родился в \n1799 году. Его отец был крепостным
крестьянином, а мать - крепостной крестьянкой. Детство и юность Пушкина прошли
в деревне Михайловское под Петербургом. В 1820-х годах семья переехала
```

mllibs
NLP assisted ML

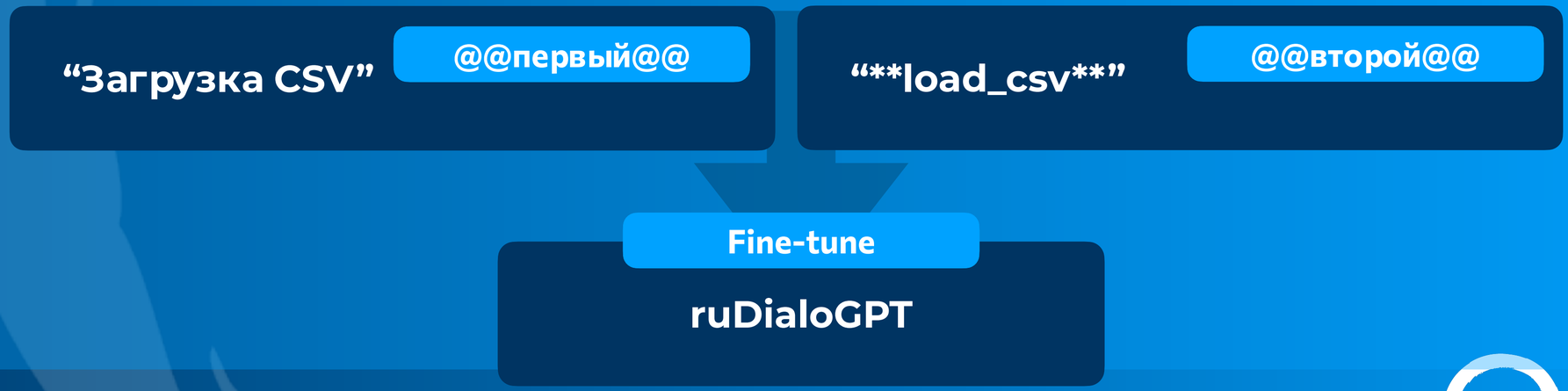
NLP подходы для вызова функции

mllibs проектная работа

NLP задачи которые могут вызвать функцию

Генерации Текста

User пишет запрос в текстовом формате и мы используем это как контекст, прогоняем через (Transformer Decoder/Seq-Seq) модель и получаем генерированный текст содержащий метку (или метки)



mllibs
NLP assisted ML

Есть связь между ФА и запросом

mllibs проектная работа

Есть NLP инструмент который может привязать вводный запрос к функции активации

Сбор библиотек/модулей

```python
from sklearn.model_selection import
train_test_split
import pandas as pd

# load the dataset
df = pd.read_csv('data.csv')

# split the dataset into training and
testing sets
X_train, X_test, y_train, y_test =
train_test_split(df.drop('target', axis=1),
df['target'], test_size=0.2, random_state=42)

print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
```

```python
import seaborn as sns

def plot_scatter(data,x,y):
    sns.scatterplot(data,x,y)
```

```python
import pandas as pd

def drop_na(data):
    data.dropna()
```

```python
import pandas as pd

def read_csv(path:str):
    pd.read_csv(path)
```

```python
import pandas as pd
import numpy as np

def drop_na(data):

    mean = np.mean(data)
    std = np.std(data)

    threshold = 3
    outlier = []
    for i in data:
        z = (i-mean)/std
        if z > threshold:
            outlier.append(i)
    print('outlier in dataset
is', outlier)
```

```python
import pandas as pd

def describe_data(data):
    data.describe()
```

```python
from sklearn.linear import
LogisticRegression

def describe_data(X,y):
    model =
LogisticRegression()
    model.fit(X,y)
```

mllibs

NLP assisted ML

Пример из 0.1.7

mllibs проектная работа

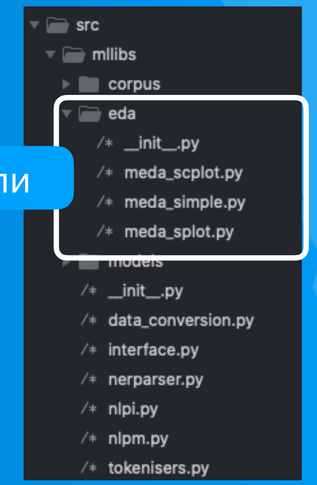
Сбор модулей для **EDA**

meda_scplot Модуль для визуализации колонок в **Seaborn**)

meda_simple Модуль для простых разведывательных операции в **Pandas**)

meda_splot Модуль для стандартных функции визуализации в **Seaborn**)

EDA модули




```python
import seaborn as sns
sns.set_theme(style="ticks")

dots = sns.load_dataset("dots")

# Define the palette as a list to specify exact values
palette = sns.color_palette("rocket_r")

# Plot the lines on two facets
sns.relplot(
    data=dots,
    x="time", y="firing_rate",
    hue="coherence", size="choice", col="align",
    kind="line", size_order=["T1", "T2"], palette=palette,
    height=5, aspect=.75, facet_kws=dict(sharex=False),
)
```

Нужны вводные данные (eg. **Dots**)

Как минимум нужно еще **x,y**

В **meda_splot** есть соответствующая функция содержащее **relplot**

mllibs

NLP assisted ML

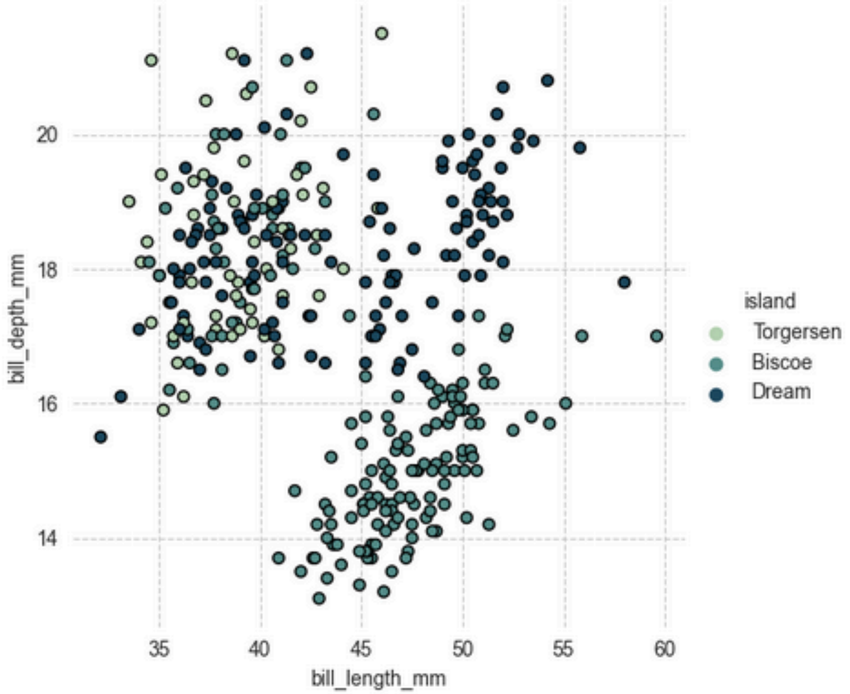
Пример из 0.1.7

mllibs проектная работа

```
'''

Seaborn Plots

'''

req = '''
create seaborn relplot
x: bill_length_mm
y: bill_depth_mm
hue: island
using penguins
'''

# interpreter['create seaborn boxplot using housing y AGE x RAD']
# interpreter['create seaborn relplot x: bill_length_mm y: bill_depth_mm hue island col=island alpha=1.0 s:50 mew: 1 using penguins']
# interpreter['create seaborn relplot x: bill_length_mm y: bill_depth_mm hue island using penguins']

interpreter.setpp({'fill':True,'alpha':1,'mew':1})
interpreter[req]
```

Запрос

```
[7]: interpreter.setpp({'fill':True,'alpha':1,'mew':1})
     interpreter[req]

plot parameter updated!
[note] NER used to clean input text!
create seaborn relplot
[note] found relevant global task [srelplot] w/ [0.91] certainty!
[note] one data source token has been set!
```



mllibs

NLP assisted ML

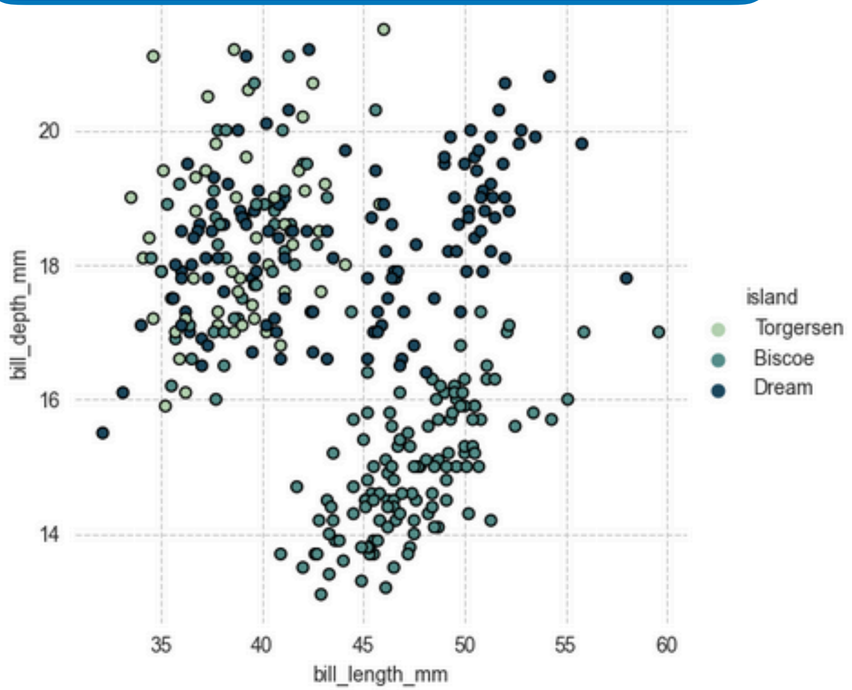
Пример из 0.1.7

mllibs проектная работа

```
'''

Seaborn Plots

'''

req = '''
create seaborn relplot
x: bill_length_mm
y: bill_depth_mm
hue: island
using penguins
'''

# interpreter['create seaborn boxplot using housing y AGE x RAD']
# interpreter['create seaborn relplot x: bill_length_mm y: bill_depth_mm hue island col=island alpha=1.0 s:50 mew: 1 using penguins']
# interpreter['create seaborn relplot x: bill_length_mm y: bill_depth_mm hue island using penguins']

interpreter.setpp({'fill':True,'alpha':1,'mew':1})
interpreter[req]
```

Фильтрация запроса
(извлекаем информацию и удаляем ненужные токены)

```
[7]: interpreter.setpp({'fill':True,'alpha
     interpreter[req]

plot parameter updated!
[note] NER used to clean input text!
create seaborn relplot
[note] found relevant global task [srelplot] w/ [0.91] certainty!
[note] one data source token has been set!
```



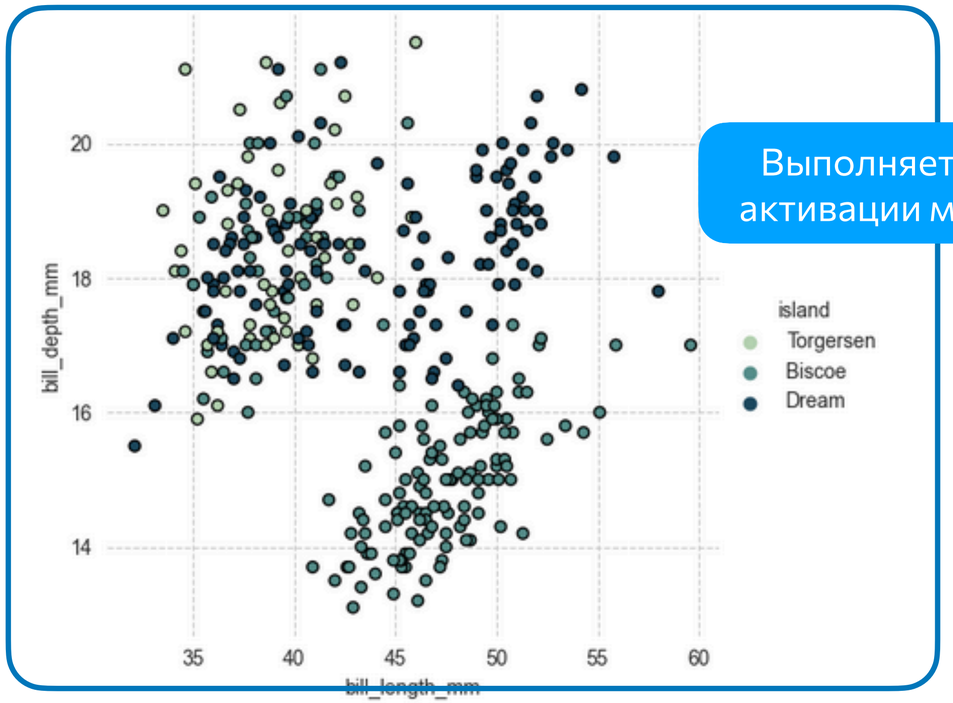
mllibs
NLP assisted ML

Пример из 0.1.7

mllibs проектная работа

```
'''

Seaborn Plots

'''

req = '''
create seaborn relplot
x: bill_length_mm
y: bill_depth_mm
hue: island
using penguins
'''

# interpreter['create seaborn boxplot using housing y AGE x RAD']
# interpreter['create seaborn relplot x: bill_length_mm y: bill_depth_mm hue island col=island alpha=1.0 s:50 mew: 1 using penguins']
# interpreter['create seaborn relplot x: bill_length_mm y: bill_depth_mm hue island using penguins']

interpreter.setpp({'fill':True,'alpha':1,'mew':1})
interpreter[req]
```

```
[7]: interpreter.setpp({'fill':True,'alpha':1,'mew':1})
     interpreter[req]

plot parameter updated!
[note] NER used to clean input text!
create seaborn relplot
[note] found relevant global task [srelplot] w/ [0.91] certainty!
[note] one data source token has been set!
```



Выполняется функция активации метки **srelplot**

mllibs

NLP assisted ML

NER в mllibs

mllibs проектная работа

```
req = '''
create seaborn relplot
x: bill_length_mm
y: bill_depth_mm
hue: island
using penguins
'''
```

Токенизатор

NER

```
'''
create seaborn relplot
'''
```

Классификатор

Функция

token_info

```
[9]: interpreter.token_info
```

[9]:

	token	index_id	type	ner_tags	data	dtype	column	ttype	ttype_storage
0	create	0	uni	O	NaN	NaN	NaN	str	create
1	seaborn	1	uni	O	NaN	NaN	NaN	str	seaborn
2	relplot	2	uni	O	NaN	NaN	NaN	str	relplot
3	x	3	uni	B-PARAM	NaN	NaN	NaN	str	x
4	:	4	uni	O	NaN	NaN	NaN	str	:
5	bill_length_mm	5	uni	O	NaN	NaN	penguins	str	bill_length_mm
6	y	6	uni	B-PARAM	NaN	NaN	NaN	str	y
7	:	7	uni	O	NaN	NaN	NaN	str	:
8	bill_depth_mm	8	uni	O	NaN	NaN	penguins	str	bill_depth_mm
9	hue	9	uni	B-PARAM	NaN	NaN	NaN	str	hue
10	:	10	uni	O	NaN	NaN	NaN	str	:
11	island	11	uni	O	NaN	NaN	penguins	str	island
12	using	12	uni	B-SOURCE	NaN	NaN	NaN	str	using
13	penguins	13	uni	O	1.0	pd.DataFrame	NaN	str	penguins

mod_args

```
# Initialise arguments dictionary
self.module_args = {'pred_task': None, 'data': None,'subset': None,
                    'splits':None,'features': None, 'target' : None,
                    'x': None, 'y': None, 'hue': None,'col':None,'row':None,
                    'col_wrap':None,'kind':'scatter', 'val':None, 'agg':None,
                    'join':'inner','axis':None,'bw':None,
                    'figsize':[None,None],'test_size':None,
                    'n_splits':None,'shuffle':None,'rs':None,
                    'threshold':None,'eps':None,'min_samples':None,'scale':None,
                    'ngram_range':None,'min_df':None,'max_df':None,
                    'tokeniser':None,'use_idf':None,'smooth_idf':None,
                    'dim':None,'window':None,
                    'epoch':None,'lr':None,'maxlen':None,'const':None,'splitter':None,
                    'neg_sample':None,'batch':None,
                    'kernel':None,'sample':None,'whiten':None,'whiten_solver':None,
                    'n_neighbours':None,'radius':None,'l1_ratio':None,
                    'alpha_1':None,'alpha_2':None,'lambda_1':None,'lambda_2':None,
                    'estimator':None,'n_estimators':None,'loss':None,
                    'criterion':None,'min_samples_leaf':None,'min_samples_split':None,
                    'max_depth':None,'max_features':None,'bootstrap':None,'oob_score':None,
                    'max_bins':None,'validation_fraction':None,'n_iter_no_change':None,
                    'nan_mode':None,'bootstrap_type':None,'l2_leaf_reg':None
                    }
```

Упрощенный

mllibs
NLP assisted ML

NER в mllibs

mllibs проектная работа

```
req = '''
create seaborn relplot
x: bill_length_mm
y: bill_depth_mm
hue: island
using penguins
'''
```

Токенизатор

NER

```
'''
create seaborn relplot
'''
```

Классификатор

Функция

token_info

```
[9]: interpreter.token_info
```

[9]:

	token	index_id	type	ner_tags	data	dtype	column	ttype	ttype_storage
0	create	0	uni	O	NaN	NaN	NaN	str	create
1	seaborn	1	uni	O	NaN	NaN	NaN	str	seaborn
2	relplot	2	uni	O	NaN	NaN	NaN	str	relplot
3	x	3	uni	B-PARAM	NaN	NaN	NaN	str	x
4	:	4	uni	O	NaN	NaN	NaN	str	:
5	bill_length_mm	5	uni	O	NaN	NaN	penguins	str	bill_length_mm
6	y	6	uni	B-PARAM	NaN	NaN	NaN	str	y
7	:	7	uni	O	NaN	NaN	NaN	str	:
8	bill_depth_mm	8	uni	O	NaN	NaN	penguins	str	bill_depth_mm
9	hue	9	uni	B-PARAM	NaN	NaN	NaN	str	hue
10	:	10	uni	O	NaN	NaN	NaN	str	:
11	island	11	uni	O	NaN	NaN	penguins	str	island
12	using	12	uni	B-SOURCE	NaN	NaN	NaN	str	using
13	penguins	13	uni	O	1.0	pd.DataFrame	NaN	str	penguins

NER

mod_args

```
# Initialise arguments dictionary
self.module_args = {'pred_task': None, 'data': None,'subset': None,
                    'splits':None,'features': None, 'target' : None,
                    'x': None, 'y': None, 'hue': None,'col':None,'row':None,
                    'col_wrap':None,'kind':'scatter', 'val':None, 'agg':None,
                    'join':'inner','axis':None,'bw':None,
                    'figsize':[None,None],'test_size':None,
                    'n_splits':None,'shuffle':None,'rs':None,
                    'threshold':None,'eps':None,'min_samples':None,'scale':None,
                    'ngram_range':None,'min_df':None,'max_df':None,
                    'tokeniser':None,'use_idf':None,'smooth_idf':None,
                    'dim':None,'window':None,
                    'epoch':None,'lr':None,'maxlen':None,'const':None,'splitter':None,
                    'neg_sample':None,'batch':None,
                    'kernel':None,'sample':None,'whiten':None,'whiten_solver':None,
                    'n_neighbours':None,'radius':None,'l1_ratio':None,
                    'alpha_1':None,'alpha_2':None,'lambda_1':None,'lambda_2':None,
                    'estimator':None,'n_estimators':None,'loss':None,
                    'criterion':None,'min_samples_leaf':None,'min_samples_split':None,
                    'max_depth':None,'max_features':None,'bootstrap':None,'oob_score':None,
                    'max_bins':None,'validation_fraction':None,'n_iter_no_change':None,
                    'nan_mode':None,'bootstrap_type':None,'l2_leaf_reg':None
                    }
```

Упрощенный

mllibs
NLP assisted ML

NER в mllibs

mllibs проектная работа

```
req = '''
create seaborn relplot
x: bill_length_mm
y: bill_depth_mm
hue: island
using penguins
'''
```

Токенизатор

NER

```
'''
create seaborn relplot
'''
```

Классификатор

Функция

token_info

```
[9]: interpreter.token_info
```

[9]:

	token	index_id	type	ner_tags	data	dtype	column	ttype	ttype_storage
0	create	0	uni	O	NaN	NaN	NaN	str	create
1	seaborn	1	uni	O	NaN	NaN	NaN	str	seaborn
2	relplot	2	uni	O	NaN	NaN	NaN	str	relplot
3	x	3	uni	B-PARAM	NaN	NaN	NaN	str	
4	:	4	uni	O	NaN	NaN	NaN	str	
5	bill_length_mm	5	uni	O	NaN	NaN	penguins	str	bill_length_mm
6	y	6	uni	B-PARAM	NaN	NaN	NaN	str	
7	:	7	uni	O	NaN	NaN	NaN	str	
8	bill_depth_mm	8	uni	O	NaN	NaN	penguins	str	bill_depth_mm
9	hue	9	uni	B-PARAM	NaN	NaN	NaN	str	hue
10	:	10	uni	O	NaN	NaN	NaN	str	
11	island	11	uni	O	NaN	NaN	penguins	str	island
12	using	12	uni	B-SOURCE	NaN	NaN	NaN	str	
13	penguins	13	uni	O	1.0	pd.DataFrame	NaN	str	

Окно поиска

Окно поиска

Окно поиска

Окно поиска

mod_args

```
# Initialise arguments dictionary
self.module_args = {'pred_task': None, 'data': None,'subset': None,
                    'split':None,'features':None,'target' : None,
                    'x': None,'y': None,'hue': None,'col':None,'row':None,
                    'col_wrap':None,'kind': 'scatter' ,'val':None, 'agg':None,
                    'join':'inner','axis':None,'bw':None,
                    'figsize':[None,None],'test_size':None,
                    'n_splits':None,'shuffle':None,'rs':None,
                    'threshold':None,'eps':None,'min_samples':None,'scale':None,
                    'ngram_range':None,'min_df':None,'max_df':None,
                    'tokeniser':None,'use_idf':None,'smooth_idf':None,
                    'dim':None,'window':None,
                    'epoch':None,'lr':None,'maxlen':None,'const':None,'splitter':None,
                    'neg_sample':None,'batch':None,
                    'kernel':None,'sample':None,'whiten':None,'whiten_solver':None,
                    'n_neighbours':None,'radius':None,'l1_ratio':None,
                    'alpha_1':None,'alpha_2':None,'lambda_1':None,'lambda_2':None,
                    'estimator':None,'n_estimators':None,'loss':None,
                    'criterion':None,'min_samples_leaf':None,'min_samples_split':None,
                    'max_depth':None,'max_features':None,'bootstrap':None,'oob_score':None,
                    'max_bins':None,'validation_fraction':None,'n_iter_no_change':None,
                    'nan_mode':None,'bootstrap_type':None,'l2_leaf_reg':None
                    }
```

Упрощенный

mllibs
NLP assisted ML

Подведем итоги

mllibs проектная работа

Какие есть подзадачи для интерпретатора

> Создания функции активации на разные темы МО

> Классификатор для подборки функции активации

> Генератор для подборки функции активации

> NER tagger для извлечения информации

> Рекомендательные системы по итогу выполнения проекта

Какие есть подзадачи для генератора

> Ответы на вопросы по темам ML

> Гиды по проектам ML

> ...

mllibs

NLP assisted ML