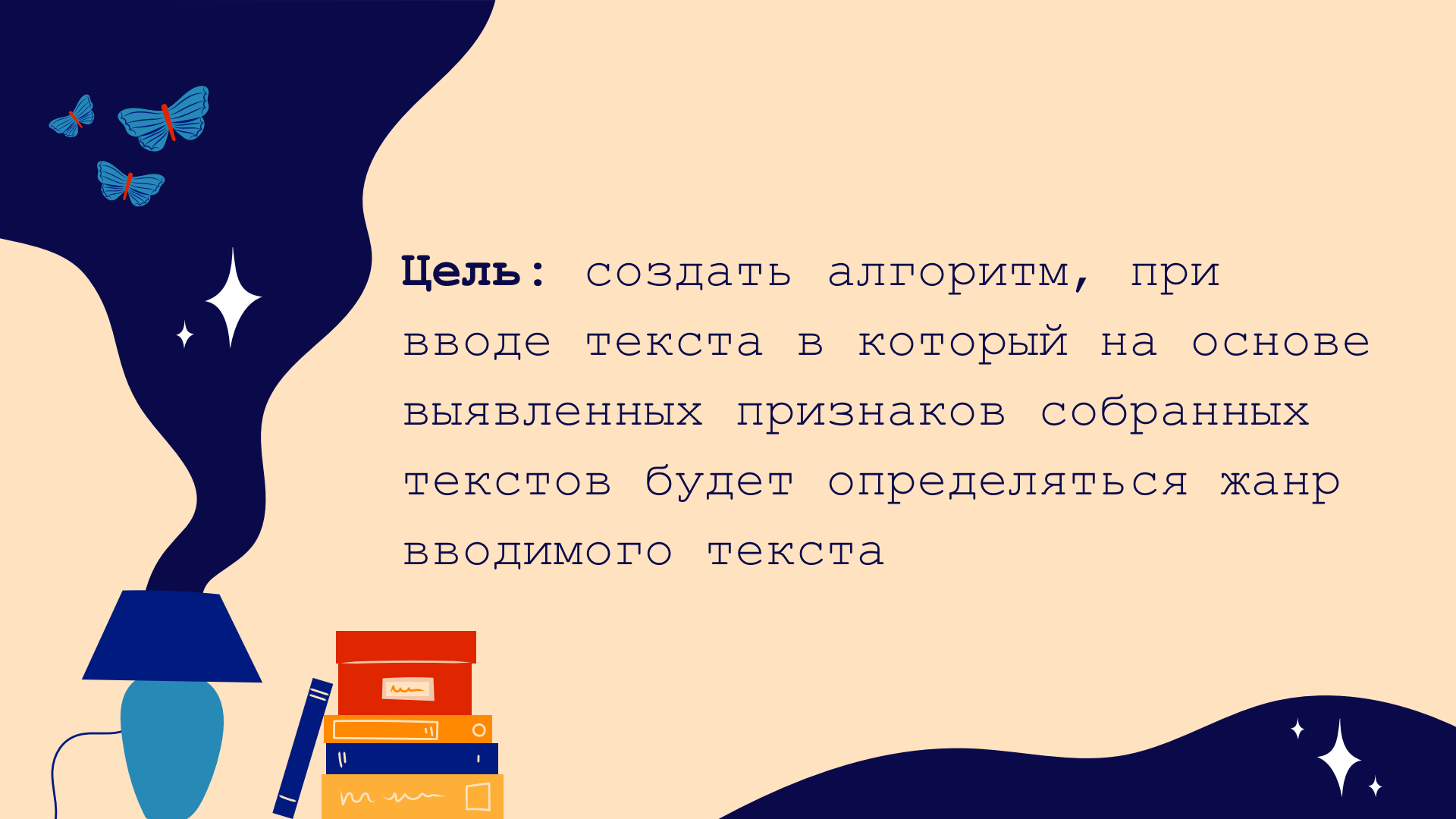


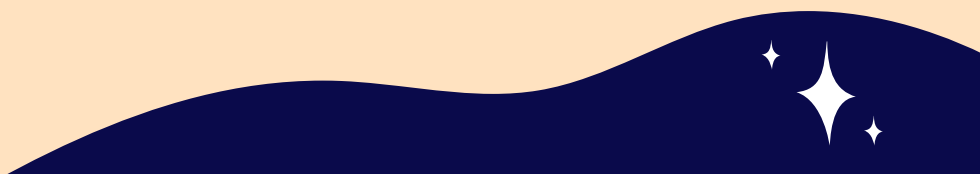
# Алгоритм для определения жанров русскоязычных текстов

Выполнили: Ким Анастасия, Лисовицкая  
Ксения, Швецова Софья





**Цель:** создать алгоритм, при вводе текста в который на основе выявленных признаков собранных текстов будет определяться жанр вводимого текста



# Этапы работы

01

Выбор текстов  
по жанрам

02

Код для анализа  
текстов

03

Презентовать  
результаты/подвести  
итоги

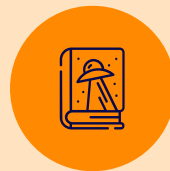


# Выбранные жанры

Художественный



Научный



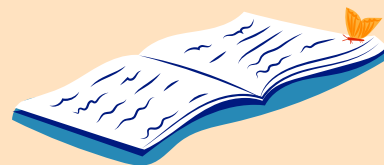
Разговорный



Публицистический



# Код



Устанавливаем нужные программы

```
!pip install pymystem3
```

```
!pip install nitk
```

```
from pymystem3
```

```
import Mystem
```

```
import re
```

# Код

Очищаем текст

```
def get_text_features(text):  
  
    text = text.lower()  
    text = text.replace('ё', 'е')  
    text = re.sub(r'^а-яё\s', ' ', text)  
    text = re.sub(r'\s+', ' ',  
text).strip()
```

Делим на слова

```
words = text.split()  
  
if not words: # Если текст пустой  
    return {  
        'avg_word_len': 0,  
        'unique_ratio': 0,  
        'long_words_ratio': 0,  
        'stopwords_ratio': 0  
    }
```

Собираем признаки

```
avg_word_len = sum(len(word) for  
word in words) / len(words) - средняя  
длина
```

```
unique_ratio = len (set(words)) /  
len(words) - кол-во неповторяющихся  
слов
```

```
long words_ratio = len([word for  
word in words if len (word) > 6])  
/ len (words) - кол-во длинных слов  
(больше 6 букв)
```

```
stopwords_ratio = len([word for  
word in words if word in  
common_words]) / len(words) - кол-во  
стоп-слов
```



# Код

Тексты-примеры:

```
training_texts = {
    'художественный': [
        "Лейтенант шел по желтому строительному
песку..."
    ],
    'Научный': [
        "Исходя из результатов эксперимента..."
    ],
    'публицистический': [
        "Невероятное открытие! Житель глухой
деревни..."
    ],
    'разговорный' : [
        "Вы когда нибудь задумывались..."
    ]
}
```

```
def predict_genre(text, genre_knowledge) :
    text_features = get_text_features (text)
    distances = {}
    for genre, genre_features in genre_knowledge.items
():
        distance = 0
        for key in text_features. keys):
            diff = abs(text_features[key] -
genre_features [key])
            distance += diff
        distances [genre] = distance
    predicted_genre = min (distances, key=distances.get)
    return predicted_genre
```

# Код

Анализ результатов:

```
print("РЕЗУЛЬТАТ АНАЛИЗА:")
print ("=" * 40)

print (f"Определенный жанр: {predicted_genre.upper()}")
print ("Насколько текст похож на каждый жанр:")

for genre, distance in distances.items ():
    similarity = max(0, 100 - distance * 20)
    print(f" {genre: 20} - {similarity: .0f}% похоже")
```

Запуск алгоритма:

```
if __name__ == "__main__":
    main()
```



Жанр	Пример	Как программа отличает
Художественный	Роман, рассказ	Больше уникальных слов
Научный	Статья, доклад	Длинные слова, меньше уникальных слов
Публицистический	Газета, блог	Много стоп-слов
Разговорный	Диалог, сообщения	Короткие слова

# Проверка алгоритма

Пьер так и не успел выбрать себе карьеры в Петербурге и действительно был выслан в Москву за буйство. История, которую рассказывали у графа Ростова, была справедлива. Пьер участвовал в связыванье квартального с медведем. Он приехал несколько дней тому назад и остановился, как всегда, в доме своего отца. Хотя он и предполагал, что история его уже известна в Москве и что дамы, окружающие его отца, всегда недоброжелательные к нему, воспользуются этим случаем, чтобы раздражить графа, он все-таки в день приезда пошел на половину отца. Войдя в гостиную, обычное местопребывание княжон, он поздоровался с дамами, сидевшими за пальцами и за книгой, которую вслух читала одна из них.

# Результат

## АЛГОРИТМ ОПРЕДЕЛЕНИЯ ЖАНРА ТЕКСТА

1. Обучаю алгоритм на примерах...

База знаний создана! Знаю 4 жанра:

- художественный
- научный
- публицистический
- разговорный

2. Введите текст для анализа:

(Можно вставить любой русский текст)

Текст для анализа:

Пьер так и не успел выбрать себе карьеры в Петербурге и действи

3. Анализирую текст...

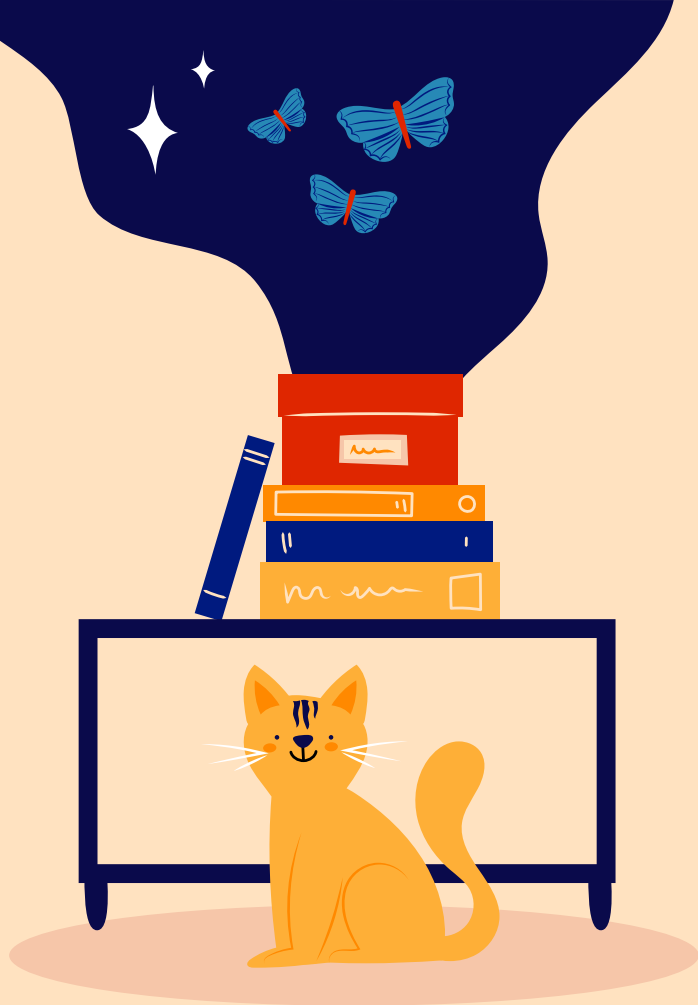
4. РЕЗУЛЬТАТ АНАЛИЗА:

Определенный жанр: ХУДОЖЕСТВЕННЫЙ

Насколько текст похож на каждый жанр:

(чем меньше число - тем больше похож)

художественный	- 0.241 (95% похоже)
разговорный	- 1.018 (80% похоже)
научный	- 1.300 (74% похоже)
публицистический	- 1.591 (68% похоже)



# Преимущества программы

Использует базовые  
лингвистические признаки

1

2

Адаптивность под  
другие жанры

3

Быстрый результат

# Ограничения программы

Простые признаки: не  
учитывают тему текста

1

2

Нет анализа смысла