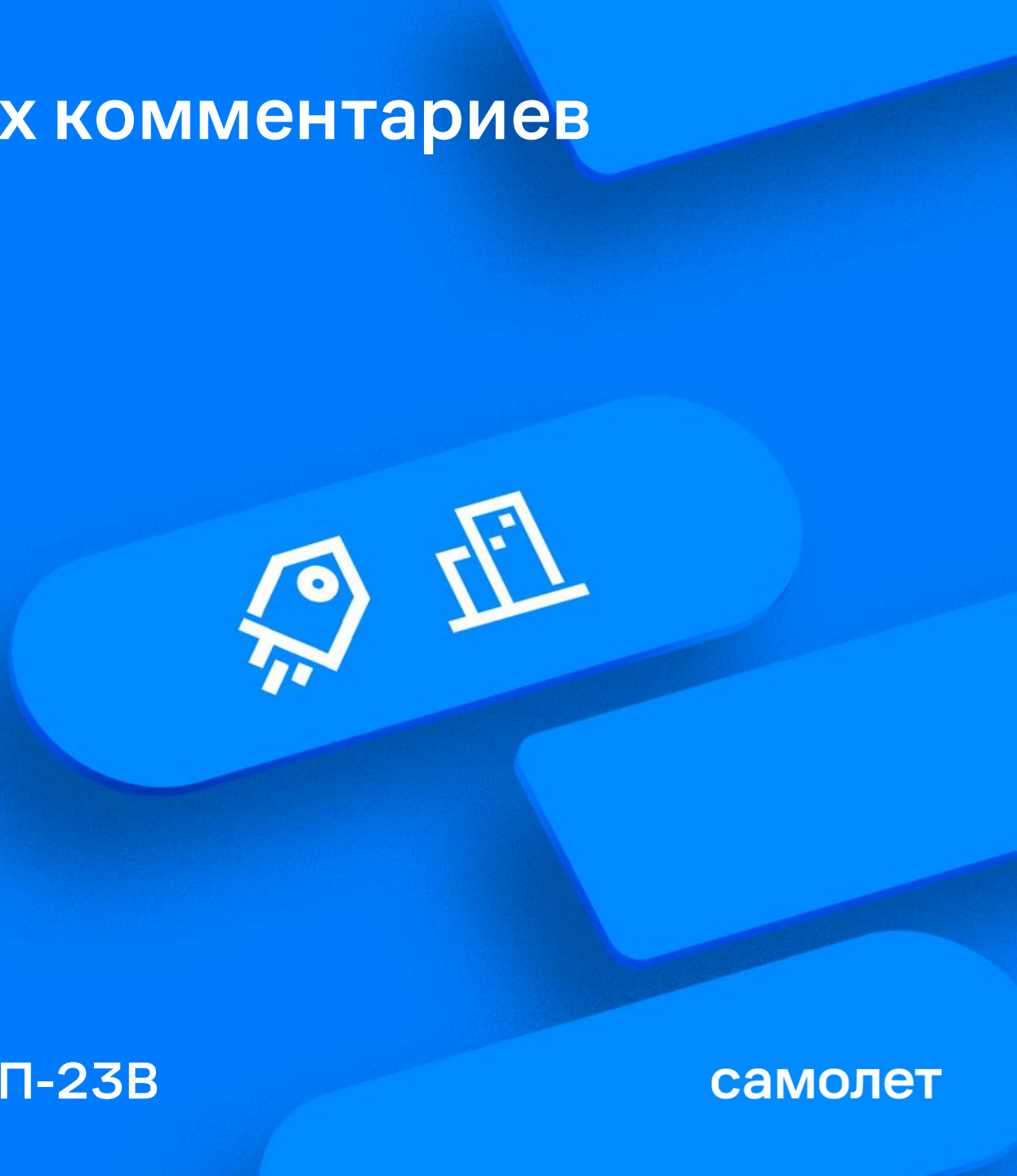
Классификация позитивных комментариев



Мурадалиев Шамиль Джамалович ИСП-23В

самолет

Постановка задачи:

Цель:

Создать высокоэффективную автоматизированную систему для определения и сортировки положительных отзывов клиентов или пользователей о деятельности управляющей компании. Основная задача состоит в том, чтобы своевременно выявлять положительные комментарии, что поможет укрепить доверие клиентов и повысить качество обслуживания.

О проекте:

Цель проекта — автоматическая идентификация положительных отзывов клиентов о деятельности управляющей компании. Задача заключается в повышении эффективности работы с отзывами, укреплении клиентского доверия и улучшении уровня предоставляемых услуг.

Шаги выполнения задачи:

Получение технического задания от компании «Самолёт»

1.

Разработка плана выполнения проекта

2.

Обработка данных: разметка данных, удаление ненужных столбцов и устранение дублирующихся записей.

3.

Представление статистических данных, включая распределение комментариев по категориям, создание облака ключевых слов и отображение количества отзывов для каждого рейтинга.

3.

Разработка модели и проведение её обучения.

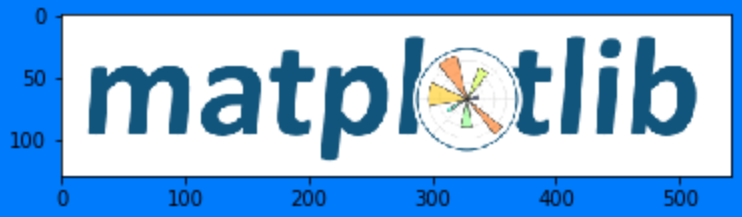
4.

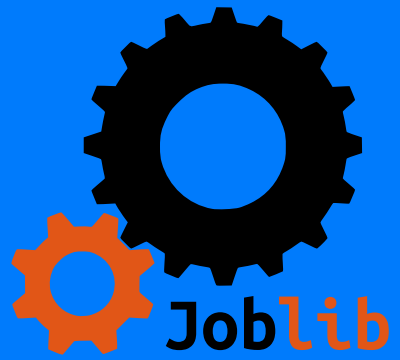
Задача выполнена

5.

самолет

Библиотеки задействованные в проекте:





- **pandas** — для работы с таблицами данных (чтение, обработка, преобразование данных).
- **json** — для парсинга данных в формате JSON, преобразования строк JSON в структуры Python.
- **matplotlib.pyplot** — для визуализации данных, создание графиков, диаграмм и изображений.
- **scikit-learn (sklearn)** — для машинного обучения: разделения данных, преобразования текста, обучения моделей, оценки и метрик.
- **numpy** — для числовых вычислений, работы с массивами и матрицами.
- **joblib** — для сохранения и загрузки обученных моделей и объектов (например, векторизаторов, энкодеров).
- **wordcloud** — для генерации облаков слов для визуализации частотных слов в текстах.

Работа с данными:

- Этот код загружает данные из файл (razmetka_komentov.csv.csv), удаляет из них ненужные колонки ('lead_time'), ('updated_at'), ('created_at'), ('annotator') и отображает первые строки таблицы.

- Это помогает подготовить данные к дальнейшему анализу, убрав лишнюю информацию.

```
df = pd.read_csv('razmetka_komentov.csv.csv')

columns_to_drop = ['lead_time', 'updated_at', 'created_at', "annotator"]
df = df.drop(columns=columns_to_drop, errors='ignore')

display(df.head())
```

	annotation_id	comment	id	rating	taxonomy	Вопрос решен	Нравится качество выполнения заявки	Нравится качество работы сотрудников	Нравится скорость отработки заявок	Понравилось выполнение заявки
0	101	спасибо	2945792	5	[{"taxonomy": [["Вопрос решен"]]}]	NaN	NaN	NaN	NaN	NaN
1	102	спасибо!	3234340	5	[{"taxonomy": [["Вопрос решен"]]}]	NaN	NaN	NaN	NaN	NaN
2	103	Отлично	3380332	5	[{"taxonomy": [["Вопрос решен"]]}]	NaN	NaN	NaN	NaN	NaN
3	104	Благодарю за оперативное решение проблемы !	3381812	5	[{"taxonomy": [["Нравится скорость отработки за...	NaN	NaN	NaN	NaN	NaN
4	105	Прекрасный специалист! Побольше таких	3461991	5	[{"taxonomy": [["Нравится качество работы сотру...	NaN	NaN	NaN	NaN	NaN

Работа с данными:

- Этот код добавляет в таблицу новые столбцы для каждой категории из списка.
- Он ищет эти категории в данных в столбце `'taxonomy'` и ставит 1, если категория есть, и 0 если нет.
- Это облегчает анализ данных.

```
categories = [
    "Нравится скорость отработки заявок",
    "Нравится качество выполнения заявки",
    "Нравится качество работы сотрудников",
    "Понравилось выполнение заявки",
    "Вопрос решен",
    "Другое"
]

for category in categories:
    if category not in df.columns:
        df[category] = 0

def extract_categories(taxonomy_str):
    try:
        data = json.loads(taxonomy_str)
        cats = []
        if isinstance(data, list):
            for item in data:
                if isinstance(item, dict) and 'taxonomy' in item:
                    nested = item['taxonomy']
                    if isinstance(nested, list):
                        for sublist in nested:
                            if isinstance(sublist, list):
                                cats.extend(sublist)
        return [cat.strip() for cat in cats]
    except:
        return []

for idx, row in df.iterrows():
    cats_in_cell = extract_categories(row['taxonomy'])
    for category in categories:
        df.at[idx, category] = 1 if category in cats_in_cell else 0

print("Обработка завершена.")
```

```
Обработка завершена.
```

самолет

Работа с данными:

- Этот код загружает данные из файла (Razmetka_Koments.csv) и для каждого из указанных столбцов проверяет, есть ли он в таблице.

- Если столбец есть, он подсчитывает и выводит, сколько в нём значений 1.

- Если столбец отсутствует, сообщает об этом.

```
df = pd.read_csv('Razmetka_Koments.csv')

columns = [
    'Нравится скорость отработки заявок',
    'Нравится качество выполнения заявки',
    'Нравится качество работы сотрудников',
    'Понравилось выполнение заявки',
    'Вопрос решен',
    'Другое'
]

for col in columns:
    if col in df.columns:
        count_ones = (df[col] == 1).sum()
        print(f"Статистика по столбцу: {col}: {count_ones}")
    else:
        print(f"Столбец '{col}' не найден в файле.")
```

```
Статистика по столбцу: Нравится скорость отработки заявок: 248
Статистика по столбцу: Нравится качество выполнения заявки: 44
Статистика по столбцу: Нравится качество работы сотрудников: 152
Статистика по столбцу: Понравилось выполнение заявки: 184
Статистика по столбцу: Вопрос решен: 245
Статистика по столбцу: Другое: 39
```

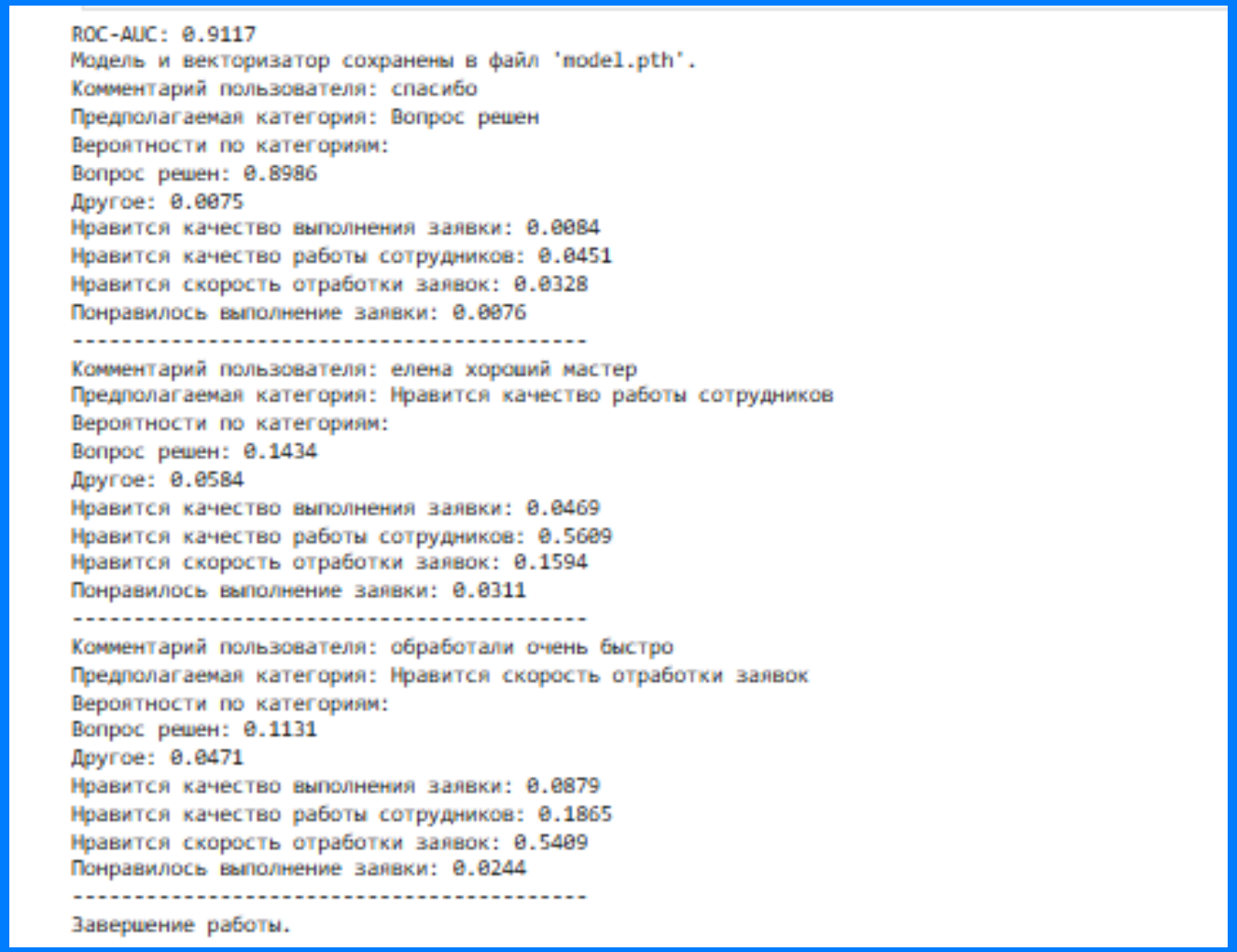
Модель:

- Модель — логистическая регрессия. Она обучается на данных, подбирая такие веса (настройки), которые позволяют максимально точно предсказывать категорию комментария на основе его слов.

- В процессе обучения модель ищет оптимальные параметры, чтобы её прогнозы были как можно более точными.

- После завершения обучения модель способна анализировать новые комментарии и примерно определять их категорию на основе наличия в них определённых слов и установленных связей между словами и категориями.

- Пример итоговой работы модели я вывел изображением на экран

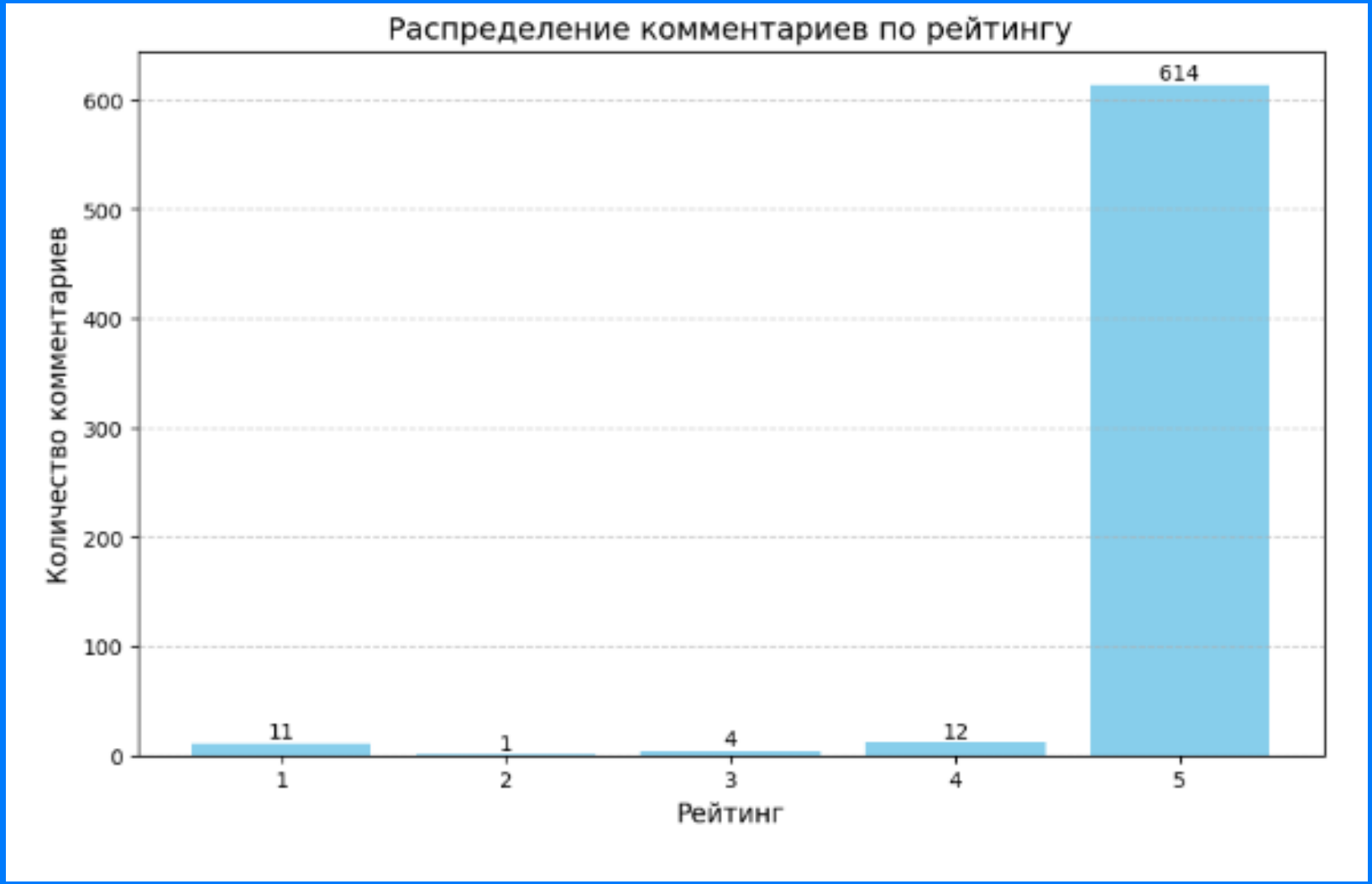


самолет

Облако слов - каких слов больше всего?

самолет

Оценки комментариев - каких оценок больше?



самолет

Итоги работы

Разработана модель классификации комментариев на основе логистической регрессии. Модель успешно предсказывает категории обратной связи (например, "Вопрос решен") и готова к интеграции в системы анализа пользовательских отзывов,

Да модель не идеальна, но она достаточно хорошо обучилась на данных которые ей предоставили

Я считаю это колоссальным опытом и развитием !!!