Введение

Данный проект посвящён исследованию пользовательских комментариев, оставленных под видео на YouTube.

Цель работы — изучить факторы, влияющие на популярность комментариев, и разработать модель, способную предсказывать количество лайков, которое может получить комментарий.

Задачи

1. Сбор данных о комментариях, видео и каналах с использованием API YouTube.
2. Обработка данных, включая вычисление доплнительных параметров, таких как длина текста или эмоциональность.
3. Исследование зависимостей популярности комментариев от длины текста, времени публикации и их эмоциональной окраски.
4. Разработка модели машинного обучения для прогнозирования популярности комментариев.

Сбор данных

Формирование списка анализируемых каналов

На первом этапе проекта необходимо было составить перечень YouTube-каналов для анализа комментариев. Основу выборки составили русскоязычные научно-популярные каналы. Список каналов был взят с сайта Pikabu и преобразован в табличный формат для удобства дальнейшей работы.

Работа включала следующие шаги:

1. Загрузка HTML-страницы со списком каналов с помощью библиотеки `requests`.
2. Извлечение названий каналов, их идентификаторов (`channel_id`) и ссылок с использованием регулярных выражений (`re`).
3. Расширение выборки вручную за счёт добавления нескольких популярных каналов.

На выходе получилась таблица с тремя колонками:

- **Название канала** (Channel name);
- **Идентификатор канала** (Channel ID);

- **Ссылка на канал** (Channel URL).

Итоговый набор данных был сохранён в файл `channels.csv` для последующего анализа.

Out[38]:

	Channel name	Channel id	
0	GEO	UCyjf5CxCNEc9ALYlalBiKDQ	https://www.youtube.com/chann
1	Prolegarium	UCnGeP_CYiOkgym9SconA2hg	https://www.youtube.com/chann
2	Skinner Show	UC2kh9KwsMmgj1LCAvwGU4HQ	https://www.youtube.com/channe
3	Utopia Show	UC8M5YVWQan_3Elm-URehz9w	https://www.youtube.com/channel/
4	Чуть-Чуть о Науке	UCKHEsjDfUOJpAev9cpjnrGg	https://www.youtube.com/chann
...	...	...	
187	Thoisoi	UCjAmQ-4NL3UZX0W_nmjn4_w	https://youtube.com/channel/UCjA
188	Доктор Грег	UC6DxE5GWRxZKwNdcOzV5hWw	https://youtube.com/channel/UC6D
189	Хауди Хо™ - Просто о мире IT!	UC7f5bVxWsm3jlZIPDzOMcAg	https://youtube.com/channel/UC
190	Вселенная Плюс	UCMrD1wosgsUpu3AE7IlljZQ	https://youtube.com/channel/UCM
191	Неземной подкаст Владимира Сурдина	UC4WAsHhtleuEGKX9x_Kbd9w	https://youtube.com/channel/UC4

192 rows × 3 columns

Сбор комментариев с каналов

После составления списка каналов начался этап сбора комментариев. Для этой задачи использовался YouTube API V3, который предоставляет доступ к метаданным и комментариям. Однако процесс был ограничен рядом факторов:

- Суточный лимит API составляет 10 000 запросов.
- Некоторые каналы оказались недоступны для сбора данных из-за отключённой функции комментирования или блокировки, что потребовало обработки ошибок.

Процесс сбора данных был организован следующим образом:

1. **Итерация по каналам.** Для каждого канала с использованием его идентификатора извлекались все связанные видео.
2. **Извлечение комментариев.** Метод `commentThreads` API использовался для получения комментариев, включая текст, количество лайков, дату публикации и ответы.
3. **Сохранение данных.** Собранные комментарии сохранялись в DataFrame. Чтобы избежать переполнения памяти, данные записывались в CSV-файл (`comments.csv`) каждые 100 запросов.

В результате было собрано более 10 миллионов комментариев, что сформировало основу для статистического анализа и помогло минимизировать влияние случайных шумов.

```
Loading comments: 0it [00:00, ?it/s]
```

Сбор метаданных о видео

Для расширения возможностей анализа были собраны метаданные о видео, включая:

- **Название видео** (Video title);
- **Дата публикации** (Video publication date);
- **Количество просмотров** (Video views).

Эти данные были сохранены в файл `videos.csv` .

На этом этапе сформирована "база данных", которая объединяет информацию о каналах, видео и комментариях. Это позволяет анализировать взаимосвязи между популярностью видео и активностью пользователей в комментариях.

Out[53]:

	Video id	Video title	Video publish date	Video views
0	KTcjP48sG3Y	САМЫЙ СТРАШНЫЙ ПЕРИОД США | ВЕЛИКАЯ ДЕПРЕССИЯ	2024-10-14T20:20:04Z	1627068
1	LNnxmBHOlOQ	История Конфуцианства - Конфуций [GEO]	2024-09-06T18:27:40Z	954871
2	E2--ncO_fhY	Фукусима: САМАЯ СТРАШНАЯ ЯДЕРНАЯ КАТАСТРОФА 21...	2024-07-23T17:49:01Z	3115484
3	eLB9JoYuj7s	ПРОРОЧЕСТВА, которые потрясли МИР! | Нострадам...	2024-05-31T17:20:00Z	4221508
4	cd9wTqHuUBw	Зодиак: Американский Задрот Убийца [Расследова...	2024-04-29T15:37:37Z	2074254
...	...	...	...	...
51956	uCRMVCyLUIg	Владимир Сурдин. ЗАГАДОЧНЫЕ находки на Венере....	2021-03-05T11:11:32Z	137136
51957	DTtS_dYSOJU	История марсоходов: от «Марс-3» до "Perseveran...	2021-02-18T21:28:27Z	49078
51958	hT08g1zreWE	На Марсе классно? Астроном Сурдин vs Noize MC....	2021-02-11T11:12:42Z	42493
51959	_ofEQNzC5A8	Сурдин: Falcon 9, Starlink и другие проекты Ма...	2021-02-08T09:51:39Z	107605
51960	h8RwMsTdpE0	Астроном Сурдин. НЕОБЫЧНЫЕ вопросы о ЛУНЕ // Н...	2021-02-04T15:10:28Z	46068

51961 rows × 4 columns

Подготовка данных для анализа

Заключительным этапом подготовки данных стало объединение всех собранных сведений в единый DataFrame. Он включал:

- **Текст комментариев** и связанные метаданные: длина текста, количество лайков, временные характеристики;
- **Эмоциональную окраску комментариев** (позитивная, нейтральная, негативная), определённую с помощью модели `Dostoevsky`;
- **Метаданные о видео**, такие как дата публикации и количество просмотров.

Объединённый датасет был сохранён в файл `alldata.csv`.

```
Warning : `load_model` does not return WordVectorModel or SupervisedModel an
y more, but a `FastText` object which is very similar.
```

Out[55]:

	Channel name	Channel id	Video title	Video id	
0	GEO	UCyjf5CxCNEc9ALYlalBiKDQ	Космическая Гонка: Почему больше НЕ ЛЕТАЮТ на ...	_P0ZK8xW8-4	8
1	GEO	UCyjf5CxCNEc9ALYlalBiKDQ	САМЫЙ СТРАШНЫЙ ПЕРИОД США | ВЕЛИКАЯ ДЕПРЕССИЯ	KTcjP48sG3Y	1
2	GEO	UCyjf5CxCNEc9ALYlalBiKDQ	Как Уничтожили Сомалийских Пиратов? [GEO]	xgQBGAiv5Ok	3
3	GEO	UCyjf5CxCNEc9ALYlalBiKDQ	ХУДШИЙ РЕЖИССЕР В ИСТОРИИ [История в Личностях]	7v5WsSoU1gw	1
4	GEO	UCyjf5CxCNEc9ALYlalBiKDQ	САМЫЙ СТРАШНЫЙ ПЕРИОД США | ВЕЛИКАЯ ДЕПРЕССИЯ	KTcjP48sG3Y	1
...	...	...	...	...	
10076128	Хауди Хо™ - Просто о мире IT!	UC7f5bVxWsm3jlZIPDzOMcAg	НОВЫЙ 100% РАБОЧИЙ ФИКС ДИСКОРД + ЮТУБ | ВОЙС ...	xjtTm3F2pWk	
10076129	Хауди Хо™ - Просто о мире IT!	UC7f5bVxWsm3jlZIPDzOMcAg	Как сделать ЛЮБОЙ сайт? За 10 минут!	r8Y0TFVVfZY	
10076130	Хауди Хо™ - Просто о мире IT!	UC7f5bVxWsm3jlZIPDzOMcAg	Как сделать ЛЮБОЙ сайт? За 10 минут!	r8Y0TFVVfZY	

	Channel name	Channel id	Video title	Video id	
10076131	Хауди Хо™ - Просто о мире IT!	UC7f5bVxWsm3jlZIPDzOMcAg	Я сделал ИИ для CS2 и она его уничтожила 3	8QWEk11UluM	3
10076132	Хауди Хо™ - Просто о мире IT!	UC7f5bVxWsm3jlZIPDzOMcAg	Я Прокачал Windows 11 потому что майкрософт не...	KEZTrTWtmLI	3

10076155 rows × 16 columns

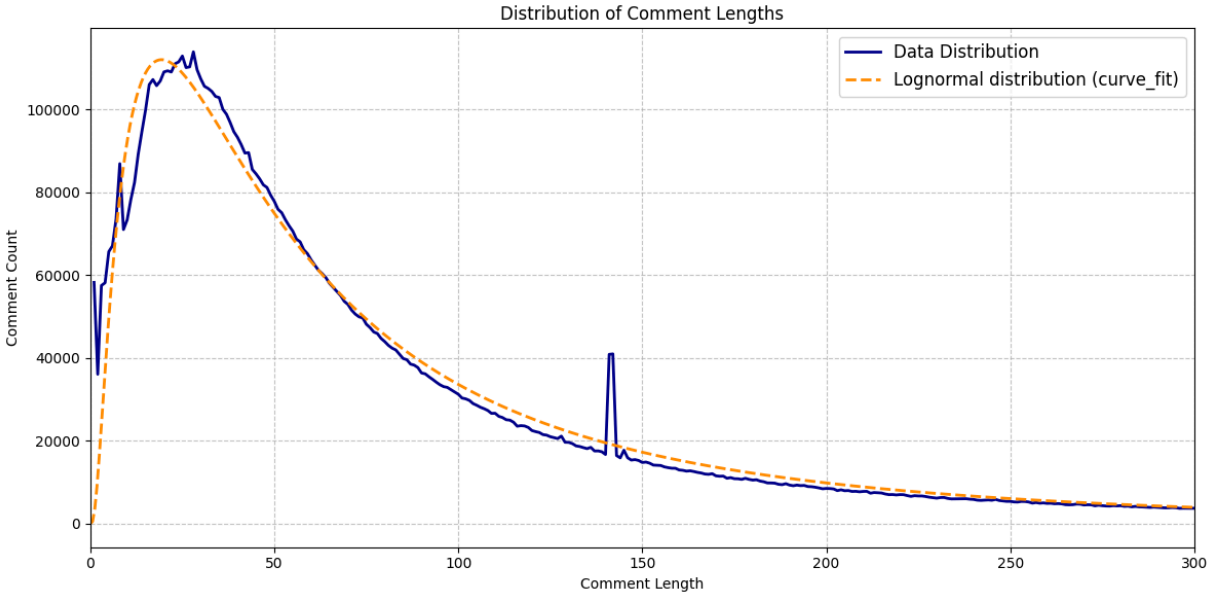
Визуализация данных и анализ комментариев

Анализ длины комментариев

На основе жизненного опыта можно предположить, что на таких платформах, как YouTube, пользователи склонны оставлять короткие комментарии. Длинные комментарии встречаются реже, что связано с тем, что их написание требует больше времени. Также логично предположить, что длина комментария может оказывать влияние на его популярность. Для проверки этих гипотез были построены графики, отображающие связь между длиной комментариев и другими параметрами.

Распределение комментариев по длине

Первым шагом было построение графика распределения количества комментариев в зависимости от их длины с помощью библиотеки `matplotlib` и написанной функции `plot_count`.



Анализ графика

График демонстрирует, что распределение комментариев по длине существенно отличается от равномерного. Наибольшее количество комментариев приходится на длину 30-40 символов. Это, вероятно, оптимальная длина, которая позволяет выразить мысль, не требуя значительных затрат времени на написание.

Из-за большого объёма данных график получился достаточно гладким, что сильнее подчёркивает некоторые выбросы. Самый заметный из них расположен в диапазоне 141-142 символов. Для дальнейшего исследования причин этой аномалии были выведены комментарии этой длины, отсортированные по количеству повторений.

Out[8]:
```
Text
Здравствуйте. Пожалуйста, посмотрите мои ролики. Не прошу подписаться! Прос
то нужно мнение со стороны. Стоит дальше делать, или я бездарность!    5780
Здравствуйте. Пожалуйста, посмотрите мои ролики. Не прошу подписаться! Прос
то нужно мнение со стороны. Стоит дальше делать, или я бездарность!    5712
Здравствуйте. Пожалуйста, посмотрите мои ролики. Не прошу подписаться! Прос
то нужно мнение со стороны. Стоит дальше делать, или я бездарность!    5680
Здравствуйте. Пожалуйста, посмотрите мои ролики. Не прошу подписаться! Прос
то нужно мнение со стороны. Стоит дальше делать, или я бездарность!    5591
Здравствуйте. Пожалуйста, посмотрите мои ролики. Не прошу подписаться! Прос
то нужно мнение со стороны. Стоит новые делать, или я бездарность!     4460
Здравствуйте. Пожалуйста, посмотрите мои ролики. Не прошу подписаться! Прос
то нужно мнение со стороны. Стоит новые делать, или я бездарность!     4441
Здравствуйте. Пожалуйста, посмотрите мои ролики. Не прошу подписаться! Прос
то нужно мнение со стороны. Стоит новые делать, или я бездарность!     4342
Здравствуйте. Пожалуйста, посмотрите мои ролики. Не прошу подписаться! Прос
то нужно мнение со стороны. Стоит новые делать, или я бездарность!     4238
Здравствуйте. Пожалуйста, посмотрите мои ролики. Не прошу подписаться! Прос
то нужно мнение со стороны. Стоит новые делать, или я бездарность!     1603
Здравствуйте. Пожалуйста, посмотрите мои ролики. Не прошу подписаться! Прос
то нужно мнение со стороны. Стоит новые делать, или я бездарность!     1599
Name: Text, dtype: int64
```

Просмотрев комментарии этой длины, стало ясно, что среди них встречаются дублирующиеся тексты, которые и вызывают такой необычный пик на графике. Например, следующий комментарий был обнаружен с различными вариациями более 40 000 раз, что сильно напоминает рекламную активность с использованием ботов:

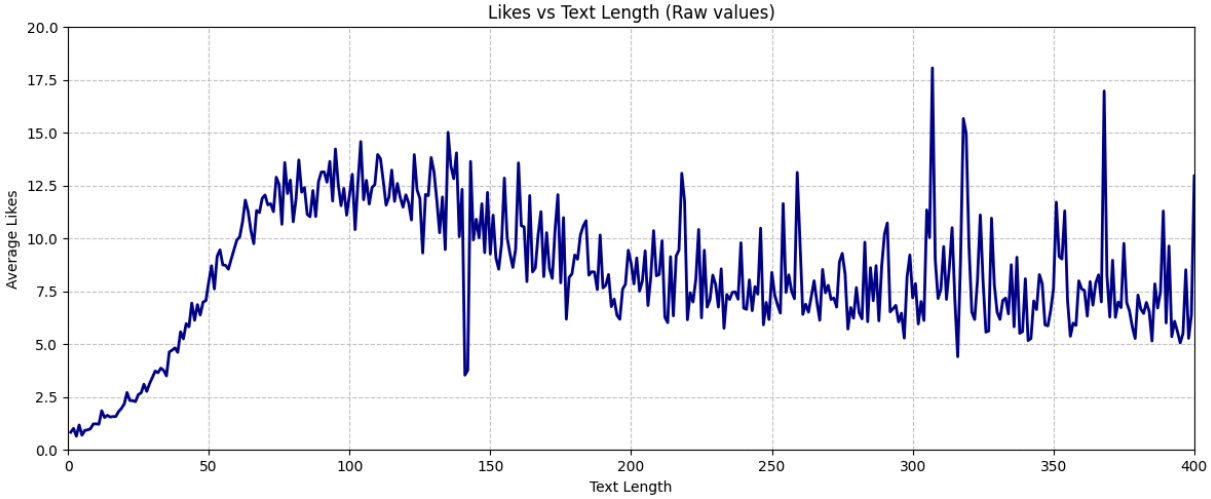
> *Здравствуйте. Пожалуйста, посмотрите мои ролики. Не прошу подписаться! Просто нужно мнение со стороны. Стоит ли продолжать делать видео, или я бездарность?*

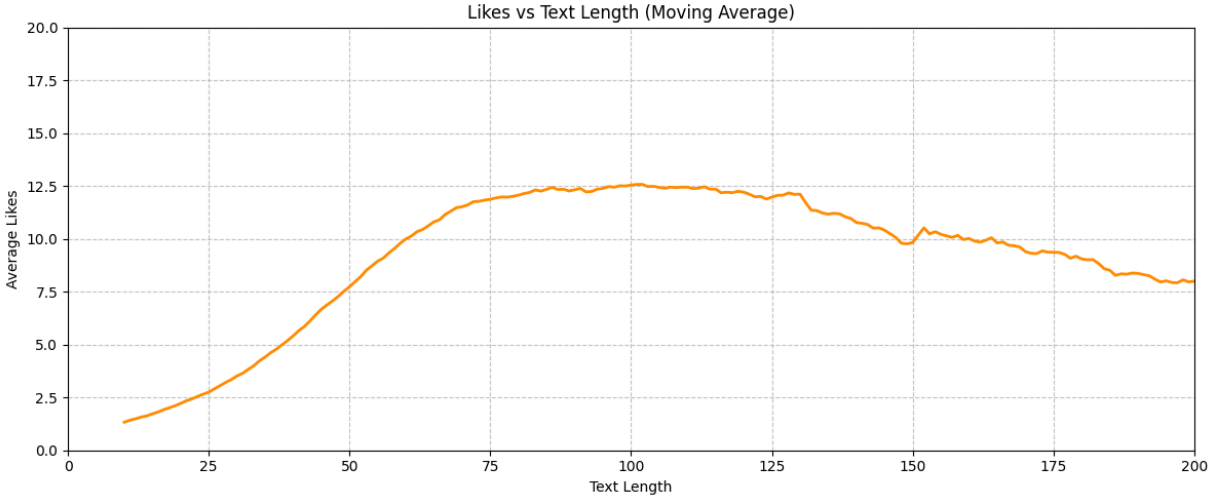
Взаимосвязь между длиной комментария и количеством лайков

Для изучения зависимости между количеством лайков и длиной комментария была написана функция `plot_likes`. С её помощью были построены графики, показывающие среднее количество лайков в зависимости от длины комментария.

Для улучшения читаемости графиков и их сглаживания применялись два подхода:

- **Удаление выбросов** с помощью функции `drop_outliers`.
- **Сглаживание с использованием скользящего среднего**, настроенного через параметр `MA_window`.

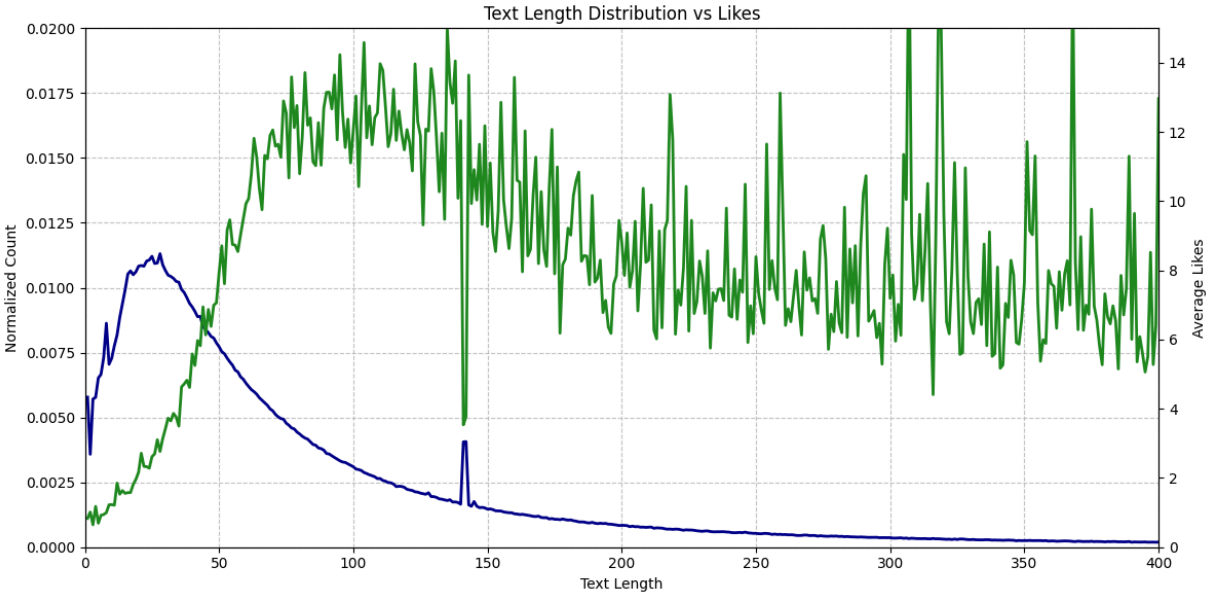




Анализ графиков

График показывает, что наибольшее количество лайков в среднем получают комментарии длиной 75-125 символов. Это объясняется тем, что короткие комментарии не содержат достаточного количества информации, чтобы заинтересовать пользователей, в то время как слишком длинные комментарии могут "отпугнуть" тех, кто не готов тратить много времени на их чтение.

На графике также можно заметить резкий спад. Для того чтобы выяснить его причину, было проведено сравнение графиков, отображающих количество комментариев и среднее количество лайков.



Анализ графика

Изучив этот график, можно сделать вывод, что причиной резкого падения лайков являются те же спам-комментарии, которые вызывают выбросы на графике количества комментариев. Действительно, никто не будет лайкать комментарий, явно написанный ботом. В совокупности с огромным количеством таких комментариев это сильно снижает среднее количество лайков.

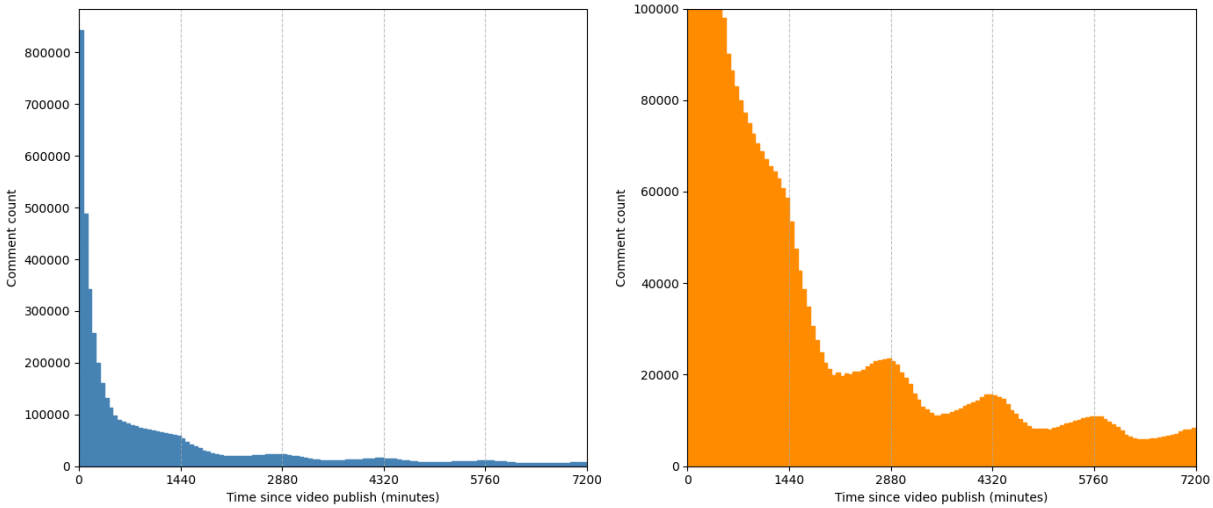
Кроме того, на графике видно, что пики количества комментариев и лайков не совпадают. Это означает, что наиболее популярные комментарии длиной 30-40 символов, вероятно, не содержат достаточно информации, чтобы быть интересными другим пользователям. Для этого требуется не менее 100 символов.

Анализ временных характеристик

Интуитивно можно предположить, что время публикации комментария также влияет на его популярность. По крайней мере, комментарий, опубликованный раньше, вероятно, будет просмотрен большим числом пользователей. Для проверки этой гипотезы были построены несколько графиков, связанных с временными характеристиками.

Распределение комментариев по времени

В первую очередь было исследовано количественное распределение комментариев в зависимости от времени, прошедшего с момента публикации видео.



Анализ графиков

На графиках видно, что основная масса комментариев оставляется пользователями в первые несколько часов после публикации видео. Это обусловлено как алгоритмами платформы, которые рекомендуют новые видео для просмотра, так и уведомлениями, поступающими подписчикам о выходе новых роликов.

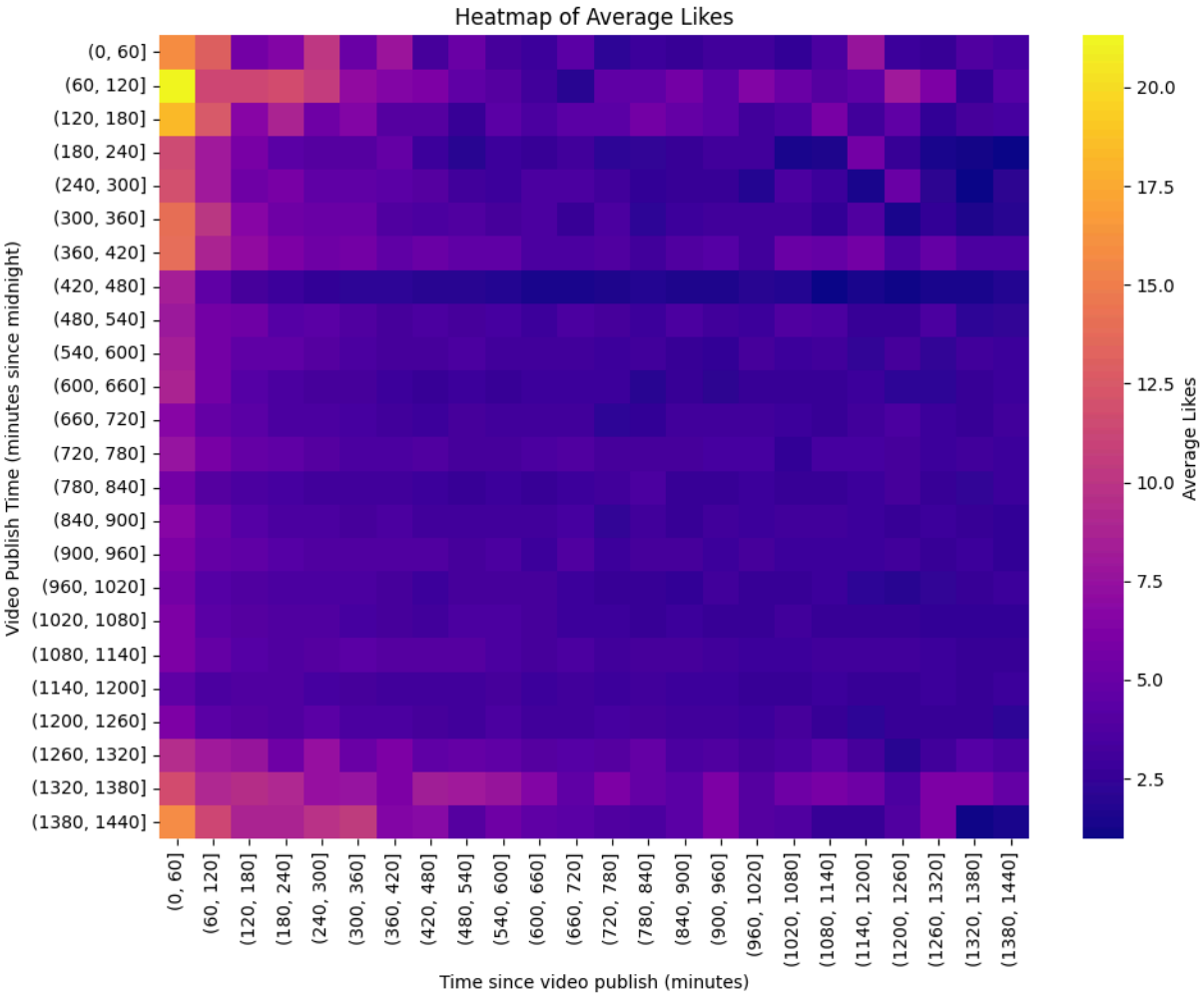
Интересно, что количество комментариев не просто постепенно уменьшается с течением времени, а имеет явно выраженные волнообразные колебания. Период этих колебаний точно равен 1440 минутам, то есть суткам. Это указывает на то, что колебания отражают ежедневные изменения в трафике платформы.

Стоит отметить, что появление этих волн на графиках стало возможным благодаря выбору русскоязычных каналов. Основная аудитория этих каналов состоит из жителей СНГ, что приводит к тому, что большинство авторов выкладывают свои видео примерно в одно и то же время. Если бы для анализа были выбраны иностранные каналы, разница в часовых поясах сгладила бы эти колебания.

Ещё одно интересное наблюдение — пики активности приходятся на 1440, 2880, 4320 и т. д. минуты. Эта закономерность подтверждает, что авторы видео действительно хорошо знают, в какое время онлайн платформы максимален, и выкладывают ролики именно в это время.

Зависимость количества лайков от времени

Далее была исследована зависимость количества лайков, получаемых комментариями, от временных характеристик. Для более глубокого анализа было учтено также время публикации видео. Рассмотрены первые сутки после публикации, и для наглядности построена тепловая карта.



Анализ графика

График подтверждает наличие зависимости между временными характеристиками и средним количеством лайков. Комментарии, оставленные в первые часы после публикации видео, получают больше лайков, что видно по градиенту, направленному слева направо. Также время публикации видео оказывает влияние: наибольшее количество лайков получают комментарии под видео, опубликованными в начале или в конце суток по UTC.

Зависимость количества лайков от года

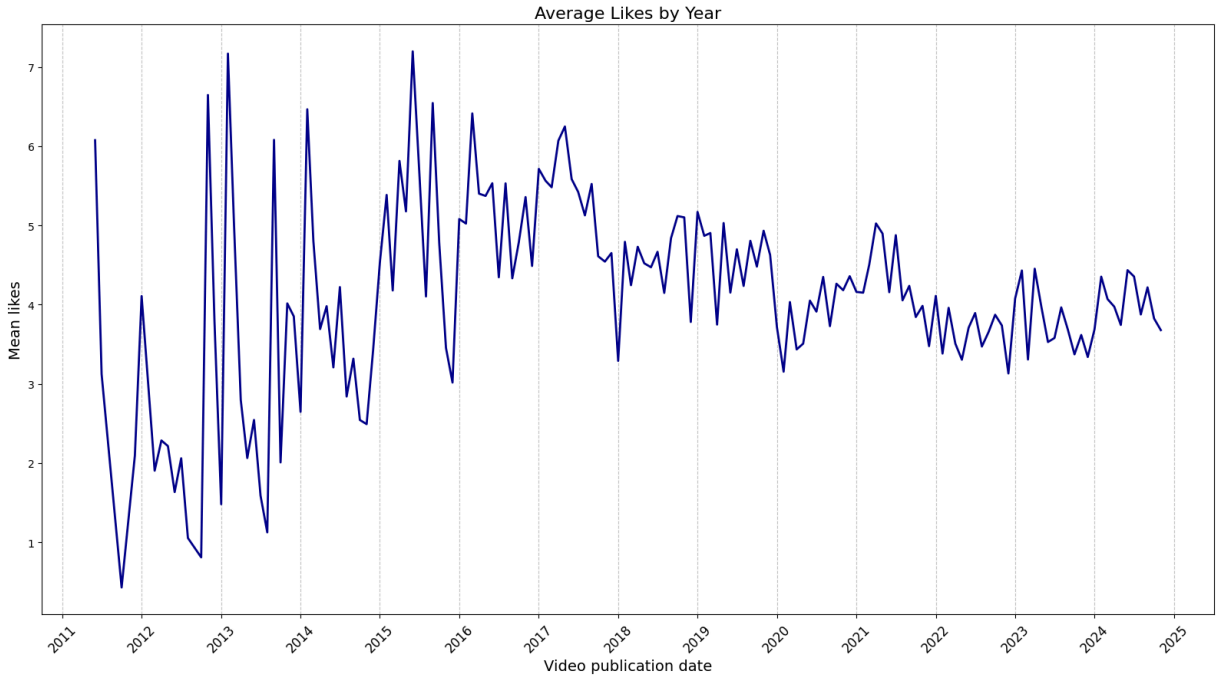
Теперь рассмотрим более широкий временной диапазон: как изменялось среднее количество лайков, получаемых комментариями, с каждым годом.

```
C:\Users\ivan\AppData\Local\Temp\ipykernel_4220\3351748260.py:2: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  plot_data["Video publication date"] = plot_data["Video publication date"].
apply(parse_iso_date)
C:\Users\ivan\AppData\Local\Temp\ipykernel_4220\3351748260.py:4: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  plot_data['Month'] = plot_data['Video publication date'].dt.to_period
('M').dt.to_timestamp()
C:\Users\ivan\AppData\Local\Temp\ipykernel_4220\3351748260.py:5: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  plot_data['Year'] = plot_data['Video publication date'].dt.to_period('Y').
dt.to_timestamp()
```



Анализ графика

На графике трудно выявить явную зависимость между годом и средним количеством лайков. В более ранние годы (2011-2017) график демонстрирует значительные колебания, вызванные ограниченным

количеством данных. С течением времени колебания становятся менее выраженными, и наблюдается стабилизация показателей на примерно одном уровне. В целом, зависимость количества лайков от года выглядит слабой и, вероятно, обусловлена внешними факторами, а не внутренними трендами.

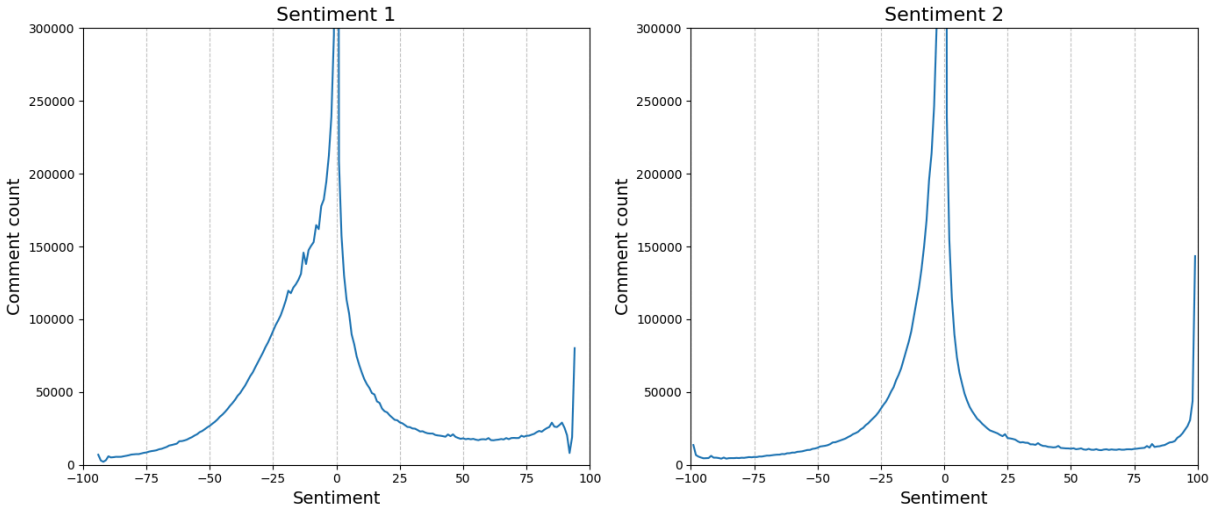
Анализ эмоциональности

Сравнение показателей

В процессе подготовки и обработки данных были использованы два варианта показателя эмоциональности комментария:

- **Sentiment 1** — логарифмическое соотношение позитивных и негативных весов
- **Sentiment 2** — разность квадратов позитивных и негативных весов

Оба показателя принимают значения в диапазоне от -100 до 100, но имеют разные функции распределения. Для их сравнения построим графики, показывающие количество комментариев в зависимости от значения каждого из показателей.



Анализ графиков

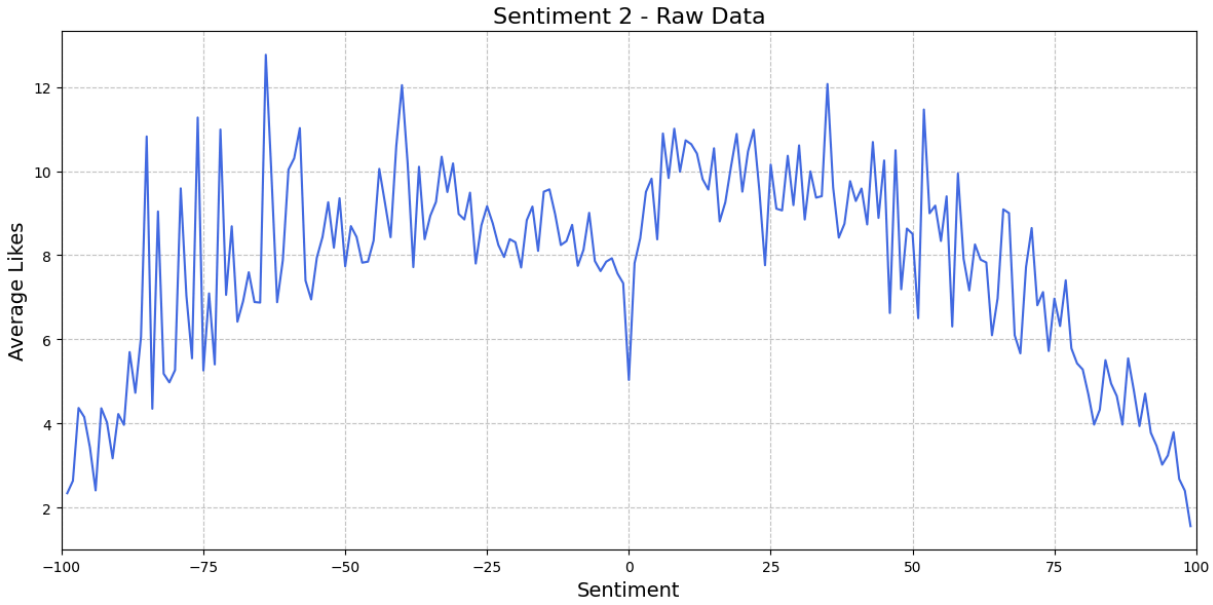
Графики имеют схожие особенности:

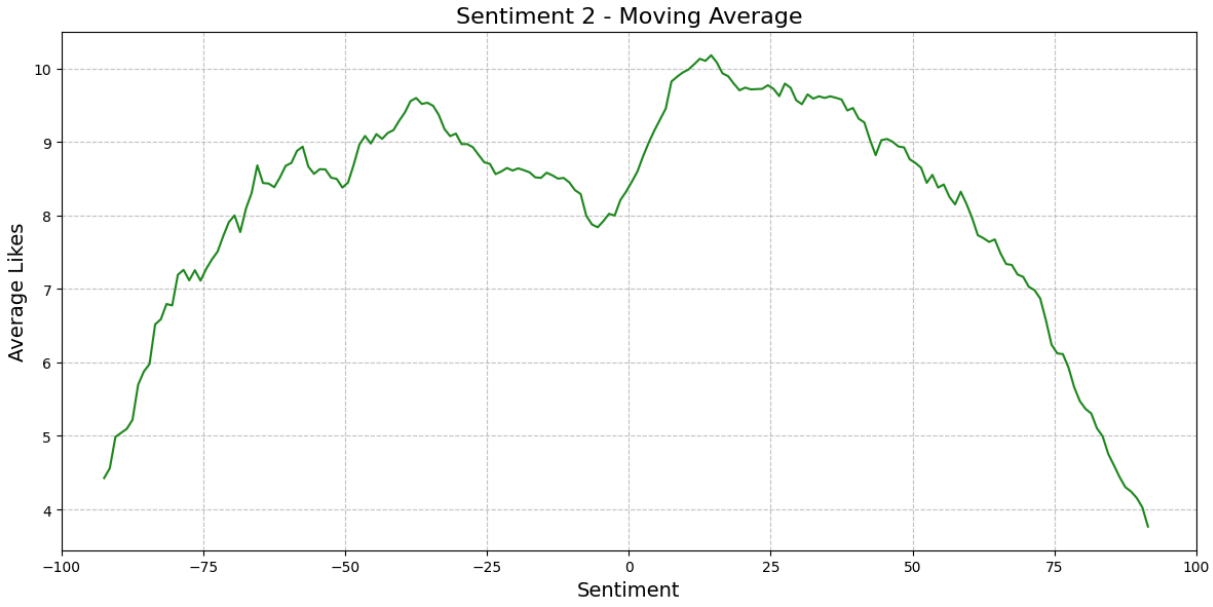
- Абсолютное большинство комментариев имеет значение, близкое к нулю.
- Имеется перекос в сторону немного негативных комментариев.
- Присутствует достаточно большое количество очень позитивных комментариев.

Однако на графике `Sentiment 1` можно заметить провалы в районе значений -90 и 90, которых нет на графике `Sentiment 2`. Эти провалы обусловлены кривизной функции на краях, когда значения положительных или отрицательных весов становятся слишком великими. В связи с этим для дальнейшего анализа в качестве показателя эмоциональности комментария будет использован `Sentiment 2`.

Зависимость количества лайков от эмоциональности

Проверим, существует ли связь между эмоциональностью комментариев и средним количеством лайков.





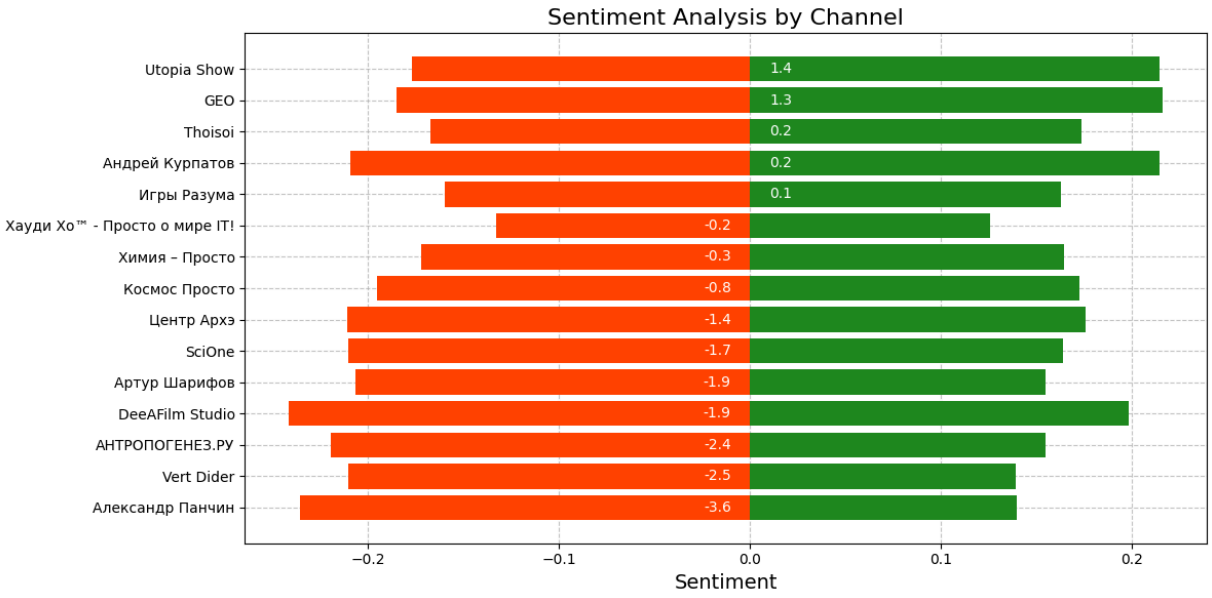
Анализ графиков

Хотя график и выглядит достаточно шумным, на нём можно заметить очевидную зависимость. Он почти симметричен относительно нуля. Комментарии с эмоциональной оценкой около нуля набирают гораздо меньше лайков. Далее, зависимость стремительно возрастает, достигает пика и затем медленно снижается к нулю по мере приближения к значению 100. Это вполне логично, ведь пользователям не интересны как чрезмерно положительные, так и сильно негативные комментарии, а также те, что не выражают ярких эмоций.

Интересно, что график не совсем симметричен: для позитивных комментариев максимальное количество лайков немного выше и ближе к нулю (~15), в то время как для негативных комментариев этот пик находится на более низком уровне (~-35).

Эмоциональность каналов

Для анализа возьмём топ-15 каналов с наибольшим количеством комментариев и сравним эмоциональность комментариев на этих каналах. Постараемся выявить, есть ли значимые различия в настроении пользователей в зависимости от канала.



Анализ графика

Как и ожидалось, средние значения эмоциональности на всех каналах близки к нулю, что объясняется присутствием как позитивных, так и негативных комментариев, зачастую в равном количестве. Тем не менее, можно заметить различия между каналами: на некоторых из них отрицательная эмоциональность в два раза выше, чем положительная.

Это может быть связано с различными факторами, такими как возрастная аудитория канала, серьезность затрагиваемых тем, провокационность контента, манера подачи материала и многие другие аспекты, влияющие на восприятие зрителей.

Обучение модели

На заключительном этапе проекта была разработана модель машинного обучения для предсказания количества лайков, которые может получить комментарий. В качестве входных данных использовались:

- **Эмбеддинги текста комментария**, полученные с помощью предобученной модели `FastText`, которые обеспечивают представление текста в виде векторных характеристик.
- **Дополнительные признаки**:
  - Длина текста комментария.
  - Эмоциональная окраска комментария.
  - Временной интервал между выходом видео и публикацией комментария.
  - Дата выхода видео, включая день недели и час выхода видео.

Архитектура модели

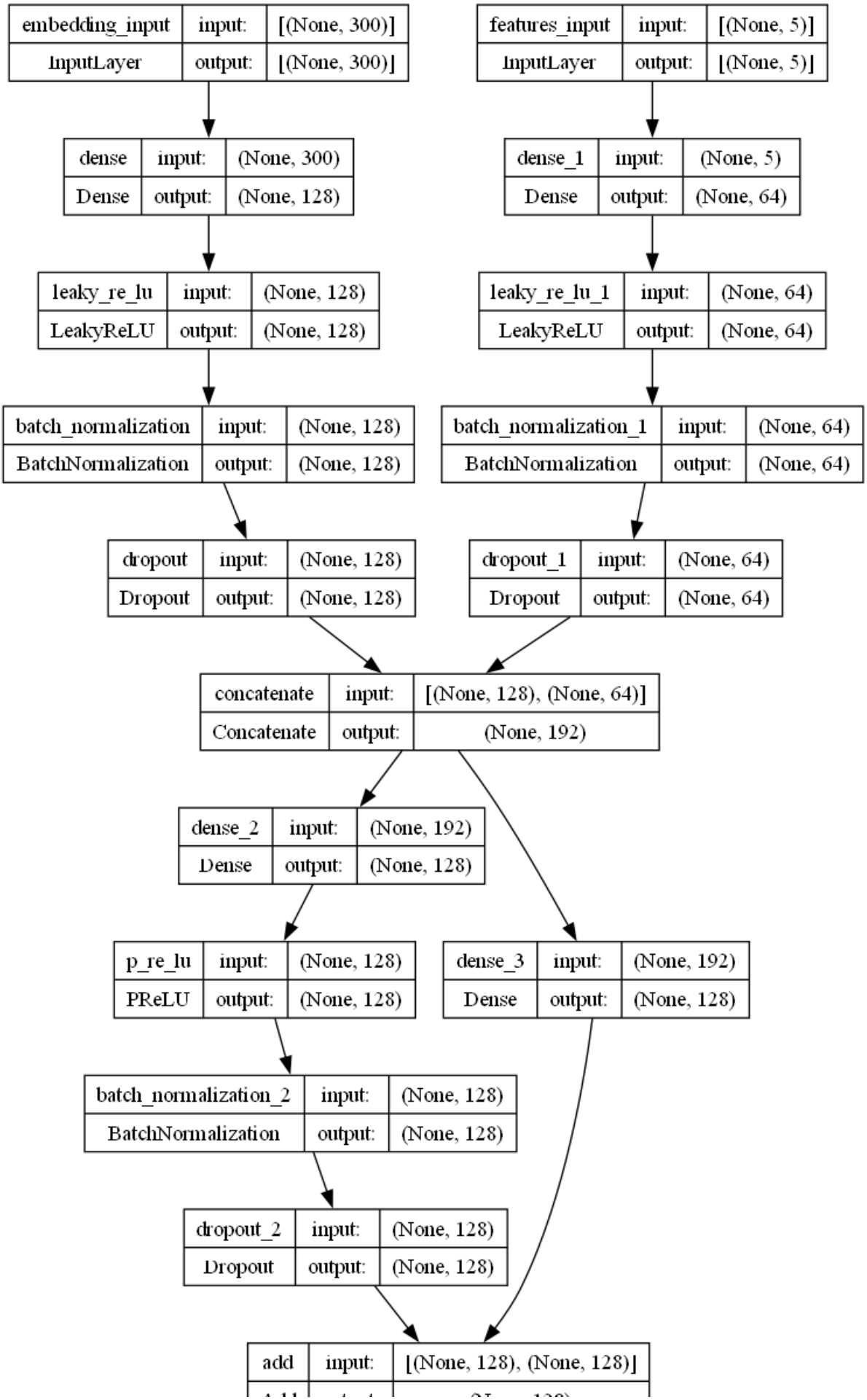
Для обработки данных была использована многоуровневая нейронная сеть, включающая следующие компоненты:

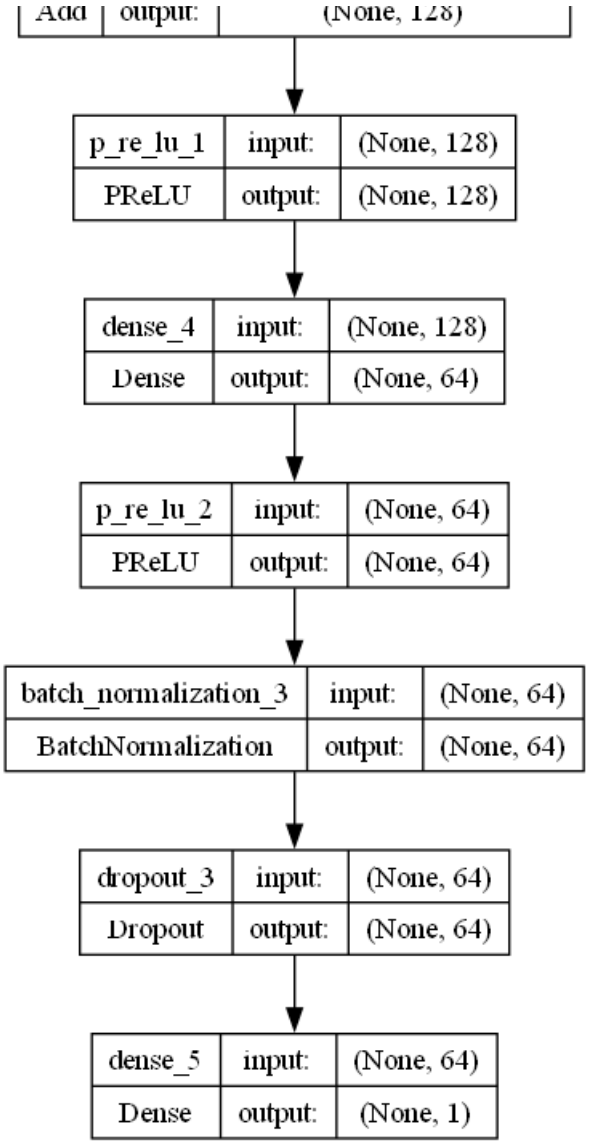
1. **Ветка для обработки эмбеддингов текста**: состояла из полносвязных слоёв с нормализацией и дропаутом для улучшения обобщающих способностей модели.

2. **Ветка для обработки дополнительных признаков**: аналогичная по структуре ветка, включающая полносвязные слои с нормализацией и дропаутом, для работы с признаками, такими как длина текста, эмоциональная окраска, временные характеристики.

Эти ветки объединялись в одной модели, после чего данные проходили через несколько полносвязных слоёв с добавлением **residual-соединений** для предотвращения переобучения.

Выходной слой представлял собой **линейный регрессор**, который прогнозировал целевую переменную — количество лайков, которое может получить комментарий.


embedding_input | input: | [(None, 300)]
InputLayer | output: | [(None, 300)]
features_input | input: | [(None, 5)]
InputLayer | output: | [(None, 5)]
dense | input: | (None, 300)
Dense | output: | (None, 128)
dense_1 | input: | (None, 5)
Dense | output: | (None, 64)
leaky_re_lu | input: | (None, 128)
LeakyReLU | output: | (None, 128)
leaky_re_lu_1 | input: | (None, 64)
LeakyReLU | output: | (None, 64)
batch_normalization | input: | (None, 128)
BatchNormalization | output: | (None, 128)
batch_normalization_1 | input: | (None, 64)
BatchNormalization | output: | (None, 64)
dropout | input: | (None, 128)
Dropout | output: | (None, 128)
dropout_1 | input: | (None, 64)
Dropout | output: | (None, 64)
concatenate | input: | [(None, 128), (None, 64)]
Concatenate | output: | (None, 192)
dense_2 | input: | (None, 192)
Dense | output: | (None, 128)
p_re_lu | input: | (None, 128)
PReLU | output: | (None, 128)
dense_3 | input: | (None, 192)
Dense | output: | (None, 128)
batch_normalization_2 | input: | (None, 128)
BatchNormalization | output: | (None, 128)
dropout_2 | input: | (None, 128)
Dropout | output: | (None, 128)
add | input: | [(None, 128), (None, 128)]



Подготовка обучающих данных

```
Warning : `load_model` does not return WordVectorModel or SupervisedModel an
y more, but a `FastText` object which is very similar.
```

```
(2000000, 300)
(2000000, 5)
(2000000,)
```

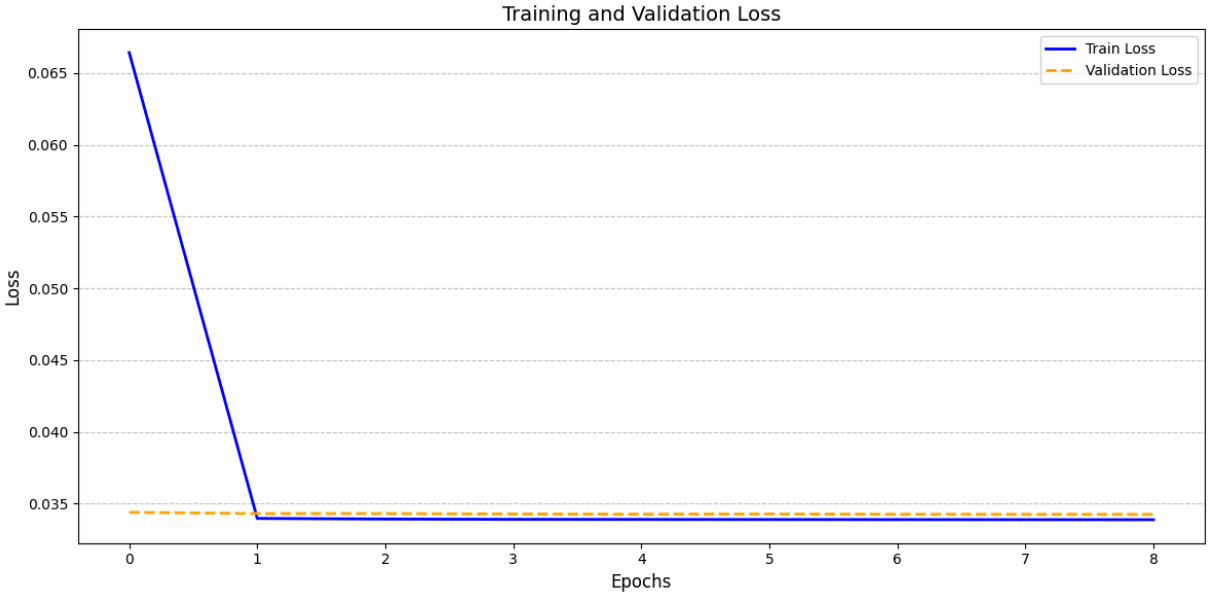
Процесс обучения

- Модель обучалась на случайной выборке из 2 000 000 комментариев.
- Для нормализации данных использовался `StandardScaler`, что обеспечивало стабильность и эффективность работы градиентного спуска.
- В качестве функции потерь была выбрана **Huber Loss**, поскольку она хорошо справляется с шумными данными и снижает влияние выбросов

на процесс обучения.

Кроме того, была применена **ранняя остановка** на основе значения ошибки на валидационной выборке (`val_loss`). Это помогло предотвратить переобучение.

```
Epoch 1/25
25000/25000 [==============================] - 82s 3ms/step - loss: 0.0664 -
mae: 0.1373 - mse: 1.0921 - val_loss: 0.0344 - val_mae: 0.0659 - val_mse: 0.
9386
Epoch 2/25
25000/25000 [==============================] - 78s 3ms/step - loss: 0.0340 -
mae: 0.0645 - mse: 1.0155 - val_loss: 0.0343 - val_mae: 0.0651 - val_mse: 0.
9381
Epoch 3/25
25000/25000 [==============================] - 78s 3ms/step - loss: 0.0339 -
mae: 0.0644 - mse: 1.0152 - val_loss: 0.0343 - val_mae: 0.0617 - val_mse: 0.
9386
Epoch 4/25
25000/25000 [==============================] - 79s 3ms/step - loss: 0.0339 -
mae: 0.0642 - mse: 1.0151 - val_loss: 0.0343 - val_mae: 0.0654 - val_mse: 0.
9379
Epoch 5/25
25000/25000 [==============================] - 79s 3ms/step - loss: 0.0339 -
mae: 0.0642 - mse: 1.0151 - val_loss: 0.0343 - val_mae: 0.0636 - val_mse: 0.
9379
Epoch 6/25
25000/25000 [==============================] - 78s 3ms/step - loss: 0.0339 -
mae: 0.0641 - mse: 1.0151 - val_loss: 0.0343 - val_mae: 0.0643 - val_mse: 0.
9381
Epoch 7/25
25000/25000 [==============================] - 78s 3ms/step - loss: 0.0339 -
mae: 0.0640 - mse: 1.0150 - val_loss: 0.0343 - val_mae: 0.0637 - val_mse: 0.
9379
Epoch 8/25
25000/25000 [==============================] - 78s 3ms/step - loss: 0.0339 -
mae: 0.0640 - mse: 1.0150 - val_loss: 0.0343 - val_mae: 0.0644 - val_mse: 0.
9379
Epoch 9/25
24992/25000 [============================>.] - ETA: 0s - loss: 0.0339 - mae:
0.0639 - mse: 1.0142Restoring model weights from the end of the best epoch:
4.
25000/25000 [==============================] - 78s 3ms/step - loss: 0.0339 -
mae: 0.0640 - mse: 1.0150 - val_loss: 0.0342 - val_mae: 0.0638 - val_mse: 0.
9378
Epoch 9: early stopping
```

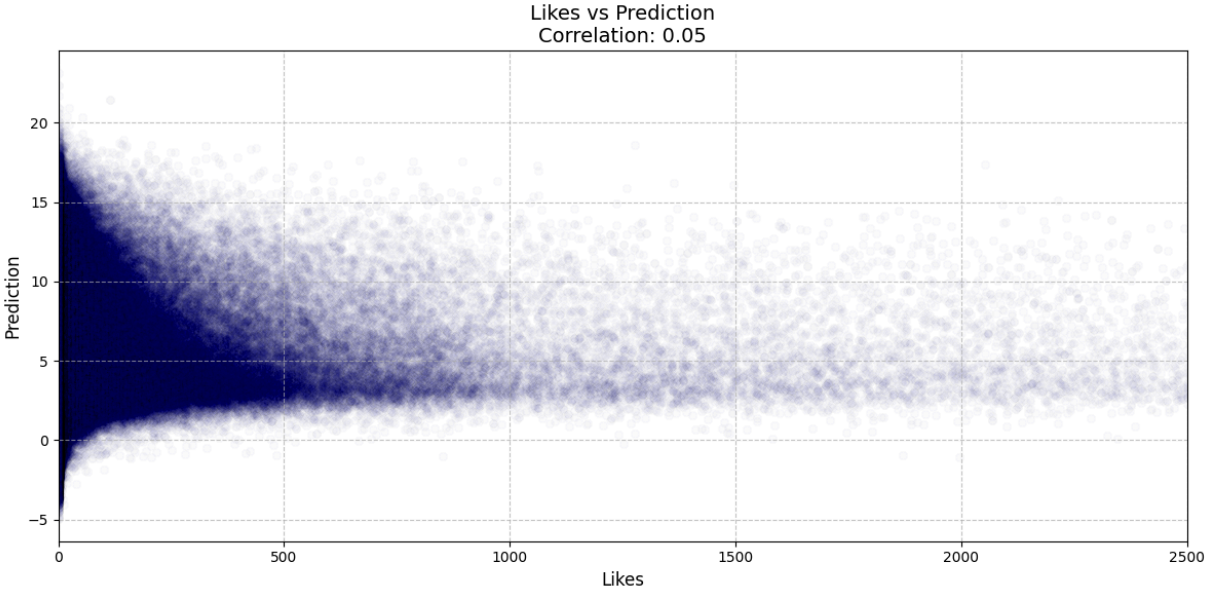


Оценка результатов обучения

После завершения обучения модель была применена ко всему датасету для вычисления коэффициента корреляции между предсказаниями и фактическими значениями (количеством лайков).

Оценка результатов обучения модели

Для оценки результатов обучения модели был вычислен коэффициент корреляции между предсказаниями и целевым значением (количеством лайков). Также был построен точечный график, который наглядно демонстрирует соответствие предсказанных и фактических значений.



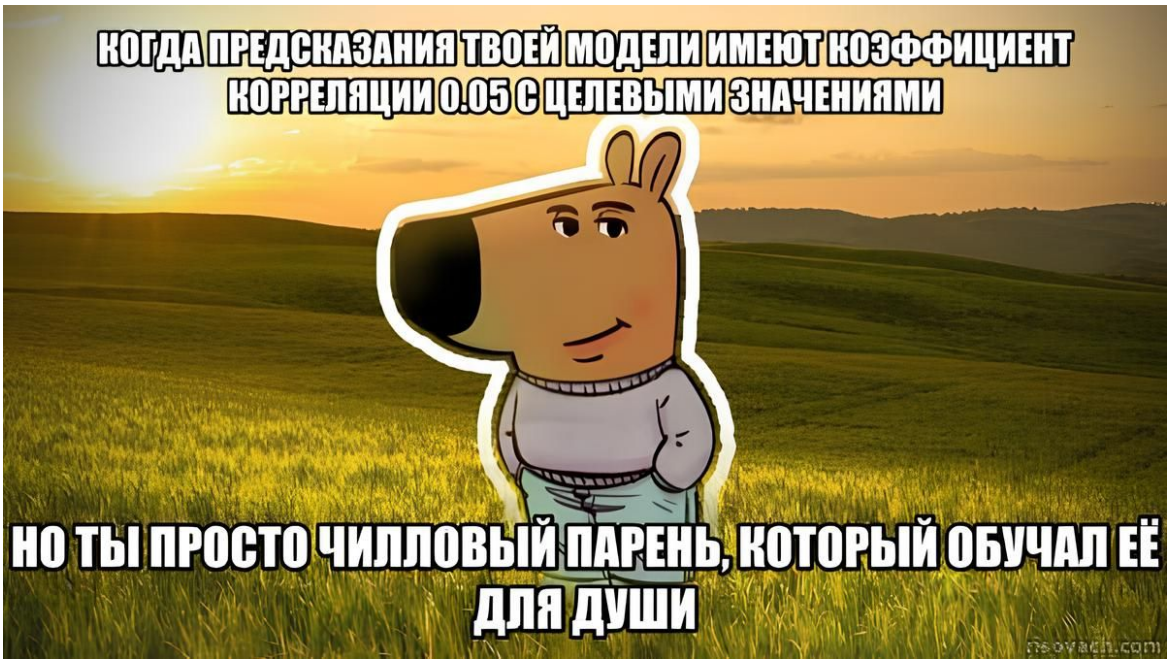
Анализ графика

Вычисленный коэффициент корреляции между предсказаниями и целевым значением равен 0.05, что указывает на крайне слабую зависимость между ними. Это означает, что модель не смогла выстроить четкую связь между текстовыми признаками и количеством лайков.

Тем не менее, на графике можно заметить, что предсказания имеют некоторую структуру, распределяясь между двумя гладкими кривыми. Однако в целом результат обучения остается неудовлетворительным. Например, если модель предсказывает количество лайков, равное 2, это число встречается как у комментариев с 0, так и с 2500 лайками, что делает такие предсказания практически бесполезными.

Вероятные причины неудачи:

- **Сложность задачи:** Количество лайков под комментариями — это крайне непредсказуемая величина, на которую влияет множество явных и неявных факторов, взаимодействующих между собой. Это делает задачу предсказания количества лайков сложной и требующей более продвинутых методов обучения для решения.
- **Неполный набор параметров:** Возможно, некоторые важные параметры не были учтены при проектировании модели. Например, содержание видео или "эффект первого лайка" (когда комментарий, уже набравший несколько лайков, имеет большую вероятность продолжать набирать лайки, в отличие от комментария без лайков).
- **Неравномерность обучающих данных:** Более 90% обучающих данных составляют комментарии с 5 и менее лайками, что делает модель склонной к предсказаниям, близким к этим низким значениям и ограничивает её способность правильно предсказывать комментарии с большим количеством лайков.



Итог проекта

В рамках проекта был проведён анализ пользовательских комментариев на платформе YouTube с целью создания модели для предсказания количества лайков под комментариями. Проект включал несколько этапов, каждый из которых позволил значительно развить технические навыки в области обработки данных и машинного обучения с использованием Python и его библиотек.

1. **Сбор и обработка данных:**

   - Для анализа было собрано более 10 миллионов комментариев, что потребовало работы с большими объёмами данных. В процессе работы активно использовались библиотеки `pandas` и `numpy` для очистки и обработки данных, таких как фильтрация, нормализация и трансформация данных в удобный формат для дальнейшего анализа.
   - Для работы с текстами комментариев применялась библиотека `FastText` для получения эмбеддингов текста, что позволило эффективно представлять текстовые данные в числовом виде для дальнейшей обработки в модели. Дополнительно использовалась библиотека `Dostoevsky` для анализа эмоциональной окраски текста с помощью алгоритмов классификации.
2. **Анализ данных:**

   - В процессе анализа данных использовались различные методы статистического анализа, а также визуализация данных с помощью библиотек `matplotlib` и `seaborn`. Для выявления закономерностей, таких как зависимость длины комментариев от их популярности и временные паттерны активности, были построены различные типы графиков и диаграмм.
   - Обнаружение аномалий, таких как спам и колебания активности, потребовало применения методов фильтрации.
3. **Разработка и обучение модели:**

   - Для создания модели был использован фреймворк `TensorFlow` с библиотекой `Keras` для построения многоуровневой нейронной сети.
   - В процессе обучения модели были применены методы, такие как нормализация данных с использованием `StandardScaler` (библиотека `scikit-learn`), использование функции потерь **Huber Loss**, которая хорошо подходит для работы с шумными данными, а

также техника ранней остановки для предотвращения переобучения.

Направления для улучшения:

1. Расширение набора признаков, включая дополнительные данные о содержании видео и социальные эффекты, что потребует дополнительных инструментов для работы с метаданными и анализа социальных сетей.
2. Использование более сложных архитектур моделей для улучшения предсказаний на основе более сложных зависимостей.

Проект предоставил ценную возможность попрактиковаться в применении инструментов Python для решения задач машинного обучения и анализа данных. Полученные навыки и знания в области обработки данных, машинного обучения и работы с большими данными будут полезны для разработки более сложных систем и моделей в будущем.