**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**(РУТ (МИИТ)**

Институт/факультет «Академия “Высшая Инженерная Школа”, АВИШ»

Специальность/Направление подготовки Информатика и вычислительная техника

**КУРСОВАЯ РАБОТА**

**по дисциплине:** Анализ больших текстовых данных и текстовый поиск

|  |  |
| --- | --- |
| **на тему:** | Построение модели классификации текста |

|  |  |
| --- | --- |
| **Студент группы ШАД 211** | **(Суворов А.С. )** |
| **Научный руководитель** | **(Проневич О. Б. )** |

**Москва 2024 г.**

# Оглавление

[**Оглавление 2**](#_heading=h.9i8no4mjwo4q)

[**Описание задачи 3**](#_heading=h.q6r9kynvnffr)

[**Решение задачи 4**](#_heading=h.8o917dkzx2ea)

[EDA 4](#_heading=h.q8p1t13t4d9d)

[Подготовка данных 4](#_heading=h.4mi7irb3ny3e)

[Построение модели 5](#_heading=h.so1w66f65vob)

[Формирование прогноза для нового сообщения 8](#_heading=h.zb2xik4bzt5z)

[Анализ качества 8](#_heading=h.nc4crbgyfjz1)

[**Выводы 9**](#_heading=h.e75yh19849jx)

[**Список использованных источников и программ 10**](#_heading=h.6r0o08d52bx)

[**Приложение 10**](#_heading=h.t0guhafb4vrx)

# Описание задачи

В варианте 23 задача заключается в разработке модели машинного обучения, способной классифицировать сообщения на основе текста, чтобы определить, из какого telegram канала они пришли (Мос. Транспорт или Авиватранспорт)

В данной курсовой работе моделируемая величина представляет собой значение, означающее, к какому чату принадлежит сообщение.

Тип задачи - классификация.

Основной критерий успеха - метрика качества модели accuracy, точность. Эта метрика была выбрана, из-за своей простоты интерпретации и способности предоставить общее представление о том, насколько хорошо модель классифицирует данные. Успехом будет считаться модель, точность которой будет больше 0.85.

# Решение задачи

## EDA

Выбранным telegram каналом с тематикой “Авиатранспорт” для сбора данных является:

* Авиаторщина | <https://t.me/aviatorshina>

Выбранным telegram каналом с тематикой “транспорт Москвы” для сбора данных является:

* Дептранс Москвы | https://t.me/DtRoad

Эти сообщества имеют достаточно большое количество подписчиков и постов.

Данные о постах в сообществах были собраны с помощью парсера который был написан самостаятельно.

Из метаданных о каждом посте были выбраны:

* текст поста, как главная категория для предсказания целевой метрики,
* дата создания поста;
* id поста.

Количество скаченных постов из Дептранс Москвы: 18983

Количество скаченных постов из Авиаторщина: 3307

В итоге EDA имеется массив данных из 3 колонок, из все три содержат текстовые данные.

## Подготовка данных

Категориальные данные (id поста) были закодированы с 0 обозначающим авиатранспорт и 1 обозначающий транспорт Москвы.

Из-за дисбаланса классов пришлось снизить количество постов от Дептранс Москвы до 6000 значений.

Затем была проведена предварительная обработка текста:

* Заполнение пропущенных значений и приведение к строковому типу: Пропущенные значения в столбце post\_text заполняются пустой строкой, а все значения приводятся к строковому типу.
* Удаление смайликов и лишних символов: С помощью регулярных выражений из текста удаляются смайлики и символы, которые не являются буквами или цифрами.
* Удаление стоп-слов: Используются стоп-слова из библиотеки NLTK для русского языка, чтобы удалить союзы и другие часто встречающиеся слова, которые не несут смысловой нагрузки.
* Удаление лишних пробелов: Любые лишние пробелы в тексте удаляются.

Подготовка данных для модели:

* Разделение данных на тексты постов (texts) и метки (labels).
* Преобразование меток в категориальный формат с использованием функции to\_categorical из библиотеки Keras.
* Токенизация текста с помощью Tokenizer из Keras, чтобы преобразовать тексты в последовательности чисел.
* Применение паддинга к последовательностям чисел с помощью pad\_sequences из Keras для обеспечения одинаковой длины последовательностей.

Разделение на обучающую и тестовую выборки было выполнено с помощью функции train\_test\_split с соотношением обучающей выборки к тестовой как 80 на 20.

## Построение модели

Для создания ансамбля моделей было необходимо использовать Python библиотеку Tensorflow.

В модели были использованы следующие слои:

* Embedding Layer (Слой вложений):
  + Этот слой используется для преобразования целых чисел (индексов слов) в векторы фиксированной размерности, называемые векторами вложений. Он позволяет модели учитывать семантические отношения между словами и их контекст.
  + Преобразует индексы слов в векторы вложений, которые затем используются как входные данные для последующих слоев.
* LSTM Layer (Слой LSTM):
  + Этот слой представляет собой рекуррентный слой, способный обрабатывать последовательные данные, сохраняя контекст и учитывая зависимости внутри последовательности.
  + Используется для обработки последовательности векторов вложений, извлечения контекстуальной информации и передачи ее следующим слоям.7
* Dropout Layer (Слой Dropout):
  + Этот слой применяется для регуляризации модели и предотвращения переобучения. Во время обучения случайно выбираются нейроны, которые будут исключены из сети.
  + Помогает уменьшить переобучение, предотвращая слишком высокую адаптацию к обучающим данным.
* Dense Layer (Полносвязный слой):
  + Назначение: Этот слой применяется для классификации или регрессии, представляя собой набор нейронов, соединенных с каждым нейроном предыдущего слоя.
  + Применение в данной модели: Используется для прогнозирования классов чатов. В этом слое используется функция активации softmax, которая преобразует выходы нейронов в вероятности принадлежности к каждому классу.

### Структура модели:

Embedding Layer:

* Входной размер: (None, 20)
* Выходной размер: (None, 20, 128)

LSTM Layer:

* Входной размер: (None, 20, 128)
* Выходной размер: (None, 20, 128)

Dropout Layer:

* Входной размер: (None, 20, 128)
* Выходной размер: (None, 20, 128)

LSTM Layer:

* Входной размер: (None, 20, 128)
* Выходной размер: (None, 128)

Dropout Layer:

* Входной размер: (None, 128)
* Выходной размер: (None, 128)

Dense Layer:

* Входной размер: (None, 128)
* Выходной размер: (None, 10)

Для обучения модели классификации, была выбрана метрика Accuracy

Полученные результаты:

Recall: 0.7795, Accuracy: 0.7423 на тренировочной выборке,

Recall: 0.9506, Accuracy: 0.6659 на валидационной выборке,

Recall: 0.9398, Accuracy: 0.5989 на тестовой выборке.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| Chat 0 | 0.97 | 0.89 | 0.93 | 531 |
| Chat 1 | 0.98 | 1.00 | 0.99 | 3732 |
| macro avg | 0.98 | 0.94 | 0.96 | 4263 |
| weighted avg | 0.98 | 0.98 | 0.98 | 4263 |

Accuracy = 0.98

Эти результаты получены при использовании тестовой выборки

## Формирование прогноза для нового сообщения

Как пример составления прогноза для новых данных можно взять новый пост, не присутствующий в датасете.

Был выбран пост из сообщества Авиаторщина о вводе новых требований для бортпроводников. Модель смогла правильно определить к какой теме относится пост.

Ссылка на пост: https://t.me/aviatorshina/4685

## Анализ качества

На рисунке 1 можно увидеть confusion matrix классификационной модели на тестовой выборке

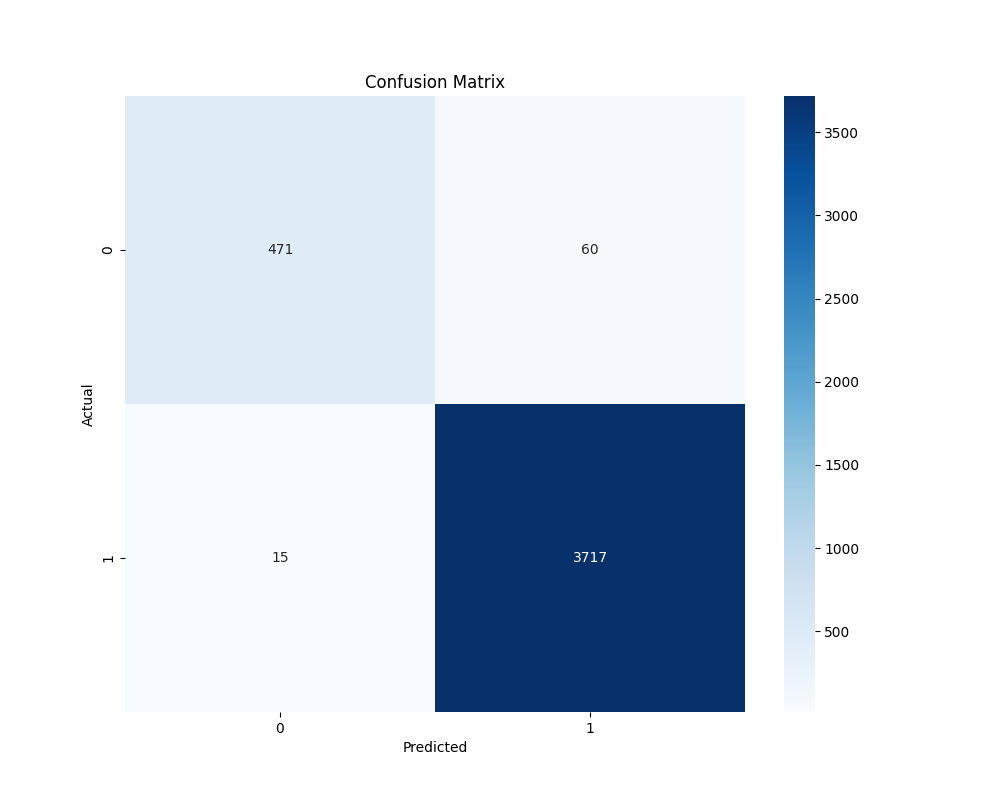


Рисунок 1

# Выводы

Используя данные из телеграм каналов о транспорте Москвы и Авиатранспорте, итоговая модель для анализа текстовых данных достиг поставленного критерия точности, что означает успех.

Анализ качества показал, что модель проявила способность прогнозировать к какому каналу относится пост.

Также в результате работы был реализован функционал определения принадлежности поста.

Для улучшения прогнозов модели можно предложить сбор большего количества данных и увеличенное количество эпох при обучении.

# Список использованных источников и программ

* Pandas
* NumPy
* Regular Expressions (re)
* Matplotlib
* Seaborn
* Scikit-learn
* TensorFlow
* Snscrape
* Csv
* NLTK

# Приложение

Код проведенной работы доступен по следующей ссылке: <https://colab.research.google.com/drive/1nFYvrl8RALUH_ReOSGD98cB0-bvbh7no?usp=sharing>