Пояснительная записка к диплому

«Классификация типов личности на основе написанного текста»

1. Выбор темы, область применения и метрики

Для того, чтобы сделать работу с людьми наиболее эффективной и привлечь новых клиентов, удержать имеющихся, нужно использовать особенные подходы. Один из вариантов: строить свою стратегию общения или представление информации в зависимости от типа личности с учетом того, что может больше понравиться и принесет большее удовлетворение человеку исходя из его психологических характеристикам. Разговор с интровертом и экстравертов разительно может отличаться.

Зная тип личности клиента/соискателя/учащегося/покупателя/пользователя, мы можем выдать ему результат, который его порадует.

Применение:

* 1. служба технической поддержки - если текст, написанный пользователем будет сразу проклассифицирован и пользователь получит метку, то сотрудник поддержки (или чат-бот) может использовать предопределенный скрипт или подход при общении с пользователем, для получения лучшей сатисфакции;
  2. на витринах продаж пользователю могут предлагаться те товары, которые в соответствии с его типом его привлекут
  3. можно выдавать персональные лендинги (в которых будет использовать персональный дизайн и предложения) в любых рекламных компаниях
  4. для HR сотрудников, предварительная диагностика на примере мотивационного письма позволит отсекать кандидатов или находить наиболее подходящих сотрудников.
  5. деление на группы в процессе обучения в любых обучающих заведениях позволит представлять программу для определенных групп в том виде, в котором она будет лучше усваиваться и приниматься.
  6. сайты знакомств могут предлагать друзей в соответствии с совместимостью типов личностей

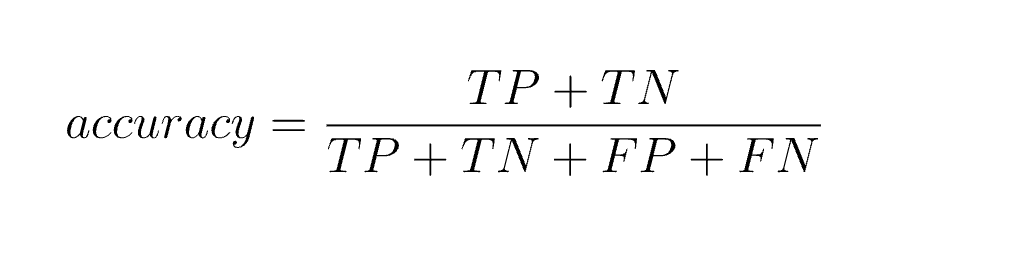
как видим применение может быть довольно широким.

Датасет для задачи был взят с Kaggle: <https://www.kaggle.com/datasets/datasnaek/mbti-type>

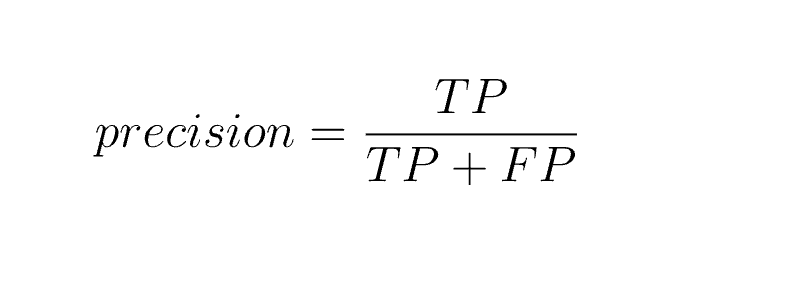
Он содержит MBTI типы - Myers–Briggs Type Indicator, эта методика была предложена двумя учеными Американки [Изабель Бриггс Майерс](https://ru.wikipedia.org/wiki/%D0%98%D0%B7%D0%B0%D0%B1%D0%B5%D0%BB%D1%8C_%D0%91%D1%80%D0%B8%D0%B3%D0%B3%D1%81_%D0%9C%D0%B0%D0%B9%D0%B5%D1%80%D1%81" \o "Изабель Бриггс Майерс) и её мать [Кэтрин Бриггс](https://en.wikipedia.org/wiki/Katharine_Cook_Briggs" \o "en:Katharine Cook Briggs) разработали свою типологию и тест на основе книги «Психологические типы» швейцарского психиатра [Карла Густава Юнга](https://ru.wikipedia.org/wiki/%D0%AE%D0%BD%D0%B3,_%D0%9A%D0%B0%D1%80%D0%BB_%D0%93%D1%83%D1%81%D1%82%D0%B0%D0%B2" \o "Юнг, Карл Густав). Методика имеет широкое применение в Европе и США.

Для оценки качества работы моделей были выбраны две метрики:

1. Accuracy (доля правильных ответов алгоритма)



2. Presision (точность) - для оценки качества работы алгоритма на каждом из классов по отдельности (с параметром average='weighted’)



Так как данные не сбалансированы, вторая метрика является более подходящей.

Хотя в ходе практики я увидела, что их значения не сильно отличаются.

1. Описание и анализ данных

Перейдем к данным моего проекта. Данные содержат 8675 строк и разделены на 16 классов согласно методике Маерс Бригс тайп индикаторс, но также по этой методике допустимо укрупнение классов по сочетанию психологических черт:

* экстраверсия (E) или интроверсия (I)
* опора на материальную информацию (S) или интуицию (N)
* логика (T) или эмоции (F)  
  четкое планирование (J) или ориентация на обстоятельства (P).

По этим признакам формируются более крупные категории и их получается всего 4.

Я проводила обучение как для 4 типов, так и для 16 классов.

Я буду называть большие категории типами, их подкатегории классами.

Аналитик (тип)

Стратег — INTJ | Ученый — INTP | Полемист — ENTP | Командир — ENTJ |

Дипломат (тип)

Активист — INFJ | Посредник — INFP | Борец — ENFP | Тренер — ENFJ |

Хранитель (тип)

Администратор — ISTJ | Консул — ESFJ | Защитник — ISFJ | Менеджер — ESTJ |

Искатель (тип)

Виртуоз — ISTP | Артист — ISFP | Делец — ESTP | Развлекатель — ESFP |

Я провела анализ постов, посмотрела их длину, распределением длин по меткам классов, посмотрела из чего состоят тексты, и какие слова чаще всего встречаются. Средняя длина поста 18-30 слов, но постов на один признак много и длина текста на признак может достигать 2000 слов и даже больше.

я решила попробовать использовать два вида типа tf-idf и w2vec и обучать как на 16, так и на 4 класса. Я использовала 5 различных алгоритмов классического ML и два варианта с глубоким обучением в дальнейшем сравнивала их результат.

1. Анализ аналогичных решений

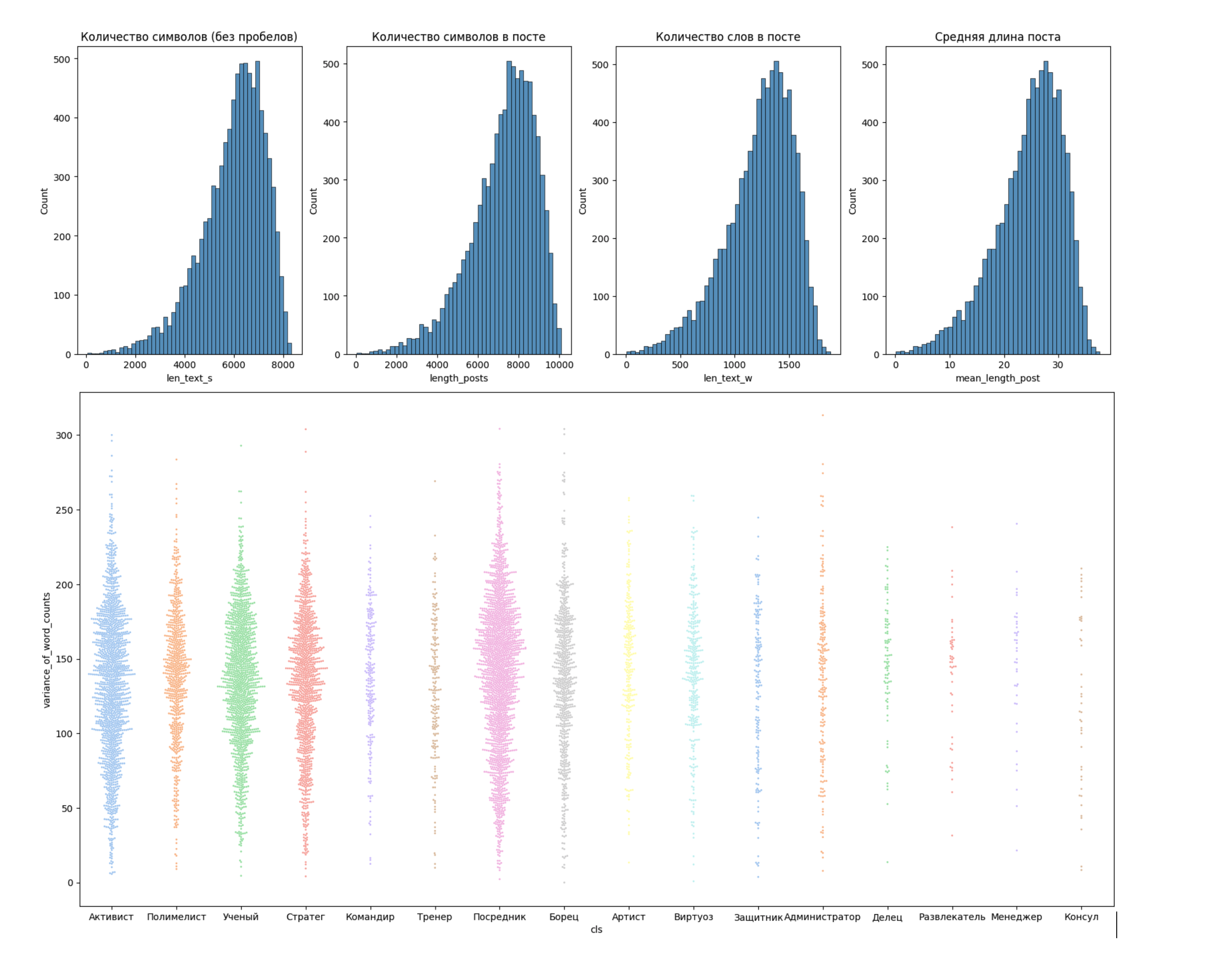
Я посмотрела какие уже есть реализованные решения, 2 решения меня заинтересовали:

**Компания MeanoTek** (Разработка нейросетей и систем ИИ) опубликовала свое решение в этой области на Хабре. К ним обратился заказчик, который попросил создать сервис на замену недоступного от IBM, суть которого заключалась в анализе текста и определял по нему ряд личностных характеристик. Их решение основано на НН Bert, у них также был несбалансированный датасет, и они постарались, чтобы их модель не зависела от языка, по крайней мере она поддерживает несколько популярных языков.

Второе решение создано специалистами по психологии и искусственному интеллекту из **НИУ ВШЭ, ОГЭУ и компании BestFitMe** (Великобритания-Россия) научили каскадную нейронную сеть определять черты личности из [«Большой пятёрки»](https://en.wikipedia.org/wiki/Big_Five_personality_traits" \t "_blank) по фотографии лица. что алгоритм делал верное заключение почти в 60% случаев, тогда как случайное угадывание обычно совпадает лишь в 50%. Превосходство в 10% кажется незначительным, однако на самом деле по точности.

1. Методика реализации задачи

Был проведен разведовательный анализ данных, по разбросу длин текстов и постов.



присутствует дисперсия длин сообщений одного типа

большее количество слов в тексте (до предобработки) на каждый признак составляет от 1000 до 1700 слов, или 6-9 тысяч символов с пробелами или 5-7 тысяч символов без учета пробелов. Длина одного поста в тексте от 18 до 32 слов.

В тексте одного признака содержится множество постов с разделителями в виде трех вертикальных черт, которые нужно было убрать

Посты содержат гиперссылки (в данном случае они бесполезны в обучении)

Посты часто содержат упоминание меток классов, которые нужно удалить, чтобы не подсказывать модели ответ.

Я вычистила символы, не являющиеся буквами, упоминания меток классов, и провела стандартные процедуры препроцессинга:

* удаление очень коротких и очень длинных слов
* очищение данных от ссылок
* удаление стоп-слов
* лемматизация

Для выравнивания дисбаланса было выполнено удаление признаков с короткими текстами (менее 100 слов).

Готовые корпуса текста в дальнейшем были преобразованы в вектора.

Я использовала два вида векторизации:

* Векторизация текстов TF-Idf (ngram\_range=(1, 1))
* Векторизация текстов word2vec (vocab\_len 23956)

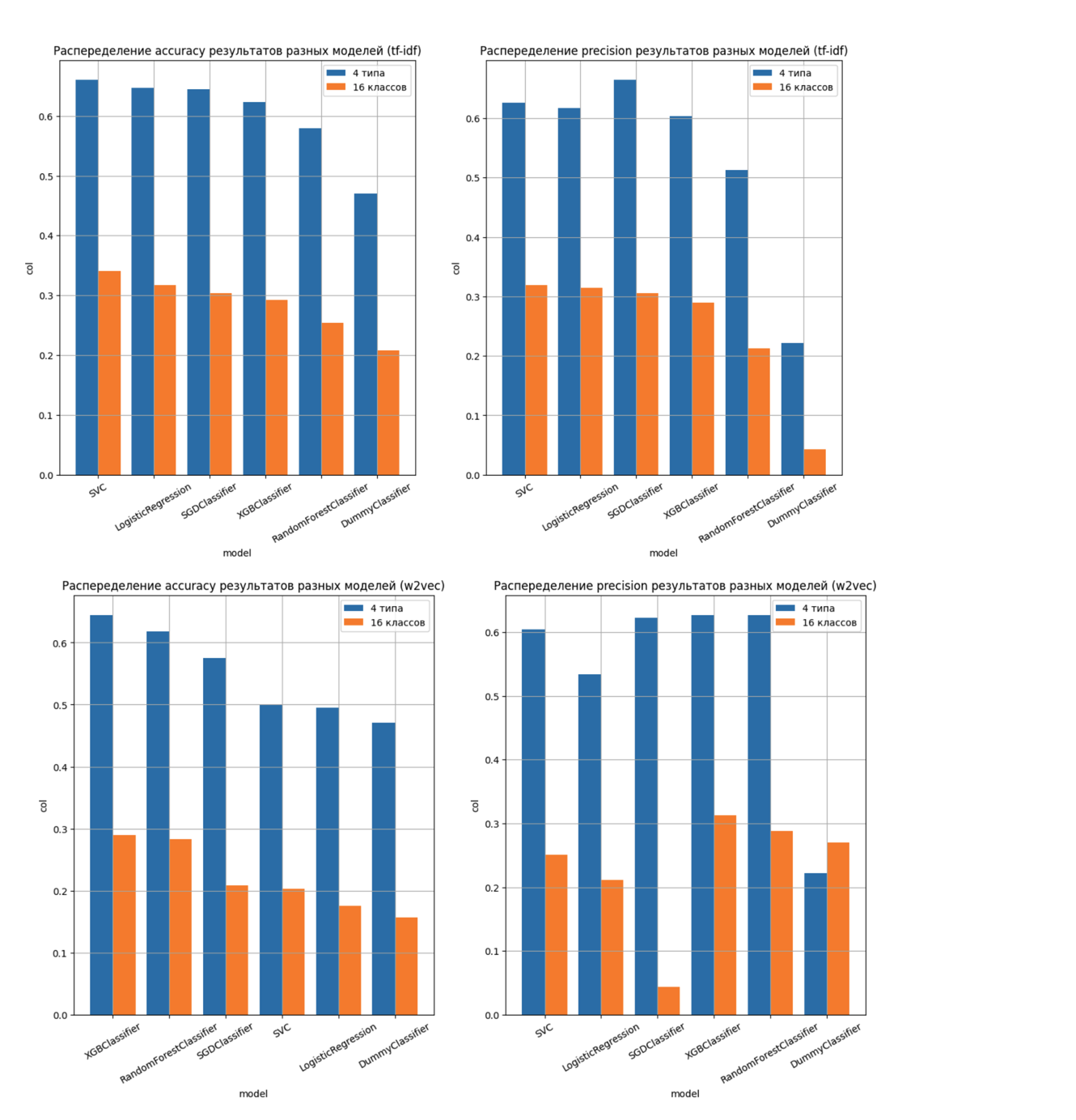
Я проводила два вида классификации:

* Классификация на 16 классов
* Классификация на 4 типа

Я обучала 5-ю различными алгоритмами (включала параметр class\_weight='balanced’ для сглаживания дисбаланса)

* RandomForestClassifier()
* LogisticRegression()
* SGDClassifier()
* SVC()
* XGBClassifier()

Таким образом я получила 40 результатов сравнения:



Для полноты исследования я решила применить глубокое обучение для своей задачи и сравнить результат с классическим ML. Здесь я провела эксперимент только для 4-х классов. Таким образом я провела обучение с помощью рекуррентной нейронной сетью, основанной на GRU и файн тьтюна Берта с использованием библиотеки

**Hugging Face.**

В рекурентную сеть я подавала эмбединги w2vec длиной – 300 с предложениями 100 токенов.

При обучении Bert – длины текстов установлены 256 (не получилось проверить с большей длиной из-за недостаточного кол-ва ресурсов) Я использовала встроенные в Hugging Face токенезаторы

from transformers import BertTokenizer,AutoTokenizer, AutoModelForMaskedLM,AutoModelForSequenceClassification

from transformers import DebertaTokenizer

from transformers import RobertaTokenizer, RobertaForSequenceClassification

print('Loading BERT tokenizer...')

tokenizer = BertTokenizer.from\_pretrained('bert-base-uncased', do\_lower\_case=True)

Модель:

The BERT model has 201 different named parameters.

==== Embedding Layer ====

bert.embeddings.word\_embeddings.weight (30522, 768)

bert.embeddings.position\_embeddings.weight (512, 768)

bert.embeddings.token\_type\_embeddings.weight (2, 768)

bert.embeddings.LayerNorm.weight (768,)

bert.embeddings.LayerNorm.bias (768,)

==== First Transformer ====

bert.encoder.layer.0.attention.self.query.weight (768, 768)

bert.encoder.layer.0.attention.self.query.bias (768,)

bert.encoder.layer.0.attention.self.key.weight (768, 768)

bert.encoder.layer.0.attention.self.key.bias (768,)

bert.encoder.layer.0.attention.self.value.weight (768, 768)

bert.encoder.layer.0.attention.self.value.bias (768,)

bert.encoder.layer.0.attention.output.dense.weight (768, 768)

bert.encoder.layer.0.attention.output.dense.bias (768,)

bert.encoder.layer.0.attention.output.LayerNorm.weight (768,)

bert.encoder.layer.0.attention.output.LayerNorm.bias (768,)

bert.encoder.layer.0.intermediate.dense.weight (3072, 768)

bert.encoder.layer.0.intermediate.dense.bias (3072,)

bert.encoder.layer.0.output.dense.weight (768, 3072)

bert.encoder.layer.0.output.dense.bias (768,)

bert.encoder.layer.0.output.LayerNorm.weight (768,)

bert.encoder.layer.0.output.LayerNorm.bias (768,)

==== Output Layer ====

bert.pooler.dense.weight (768, 768)

bert.pooler.dense.bias (768,)

classifier.weight (4, 768)

classifier.bias (4,)

Параметры для рекрентной сети GRU :

batch\_size = 16

criterion = nn.CrossEntropyLoss()

optimizer = torch.optim.Adam(model.parameters(), lr=1e-2)

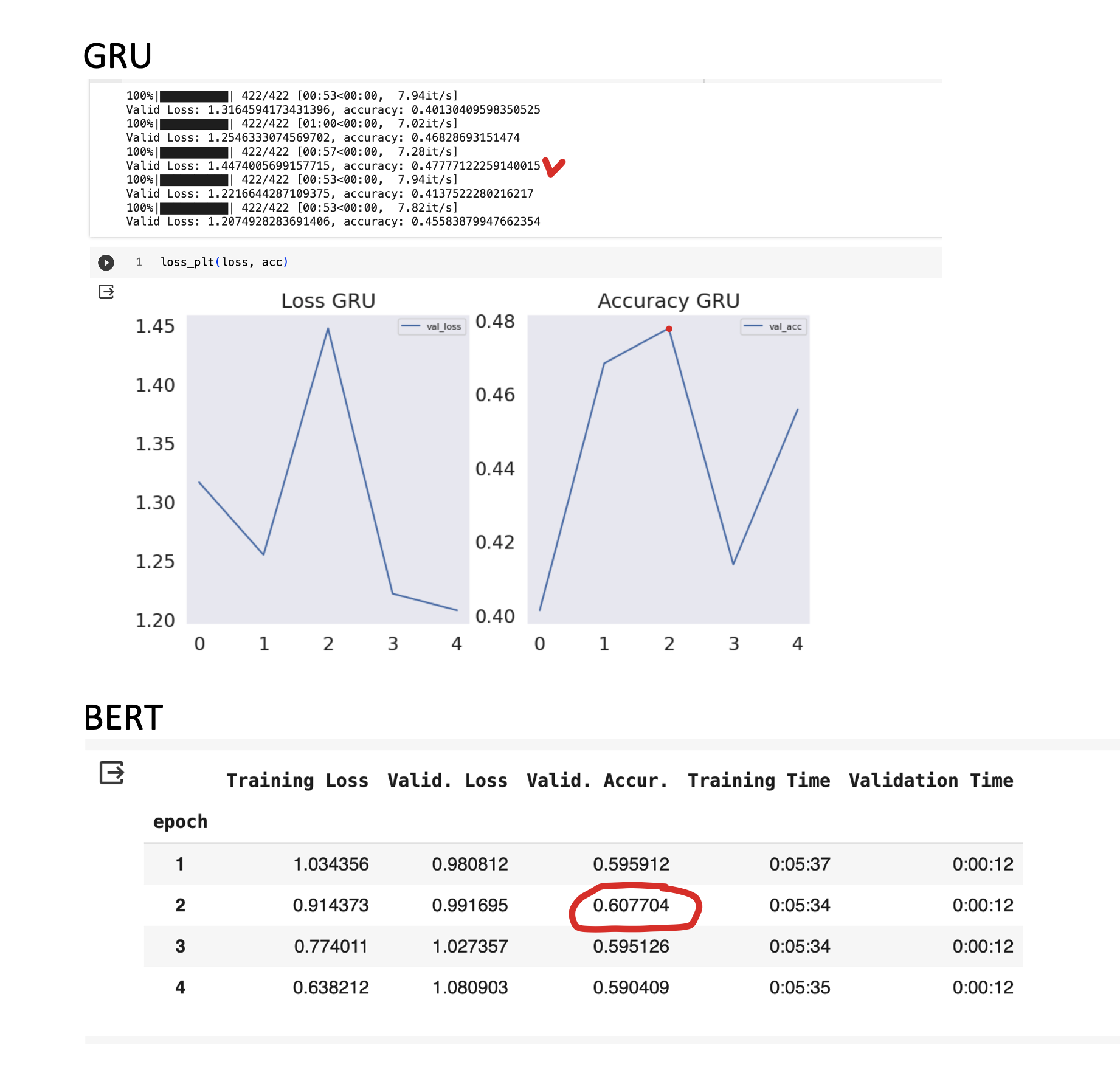
num\_epochs = 5

FineTune Bert

(batch\_size = 16, epochs = 4,

AdamW( lr = 0.00002 + get\_linear\_schedule\_with\_warmup),

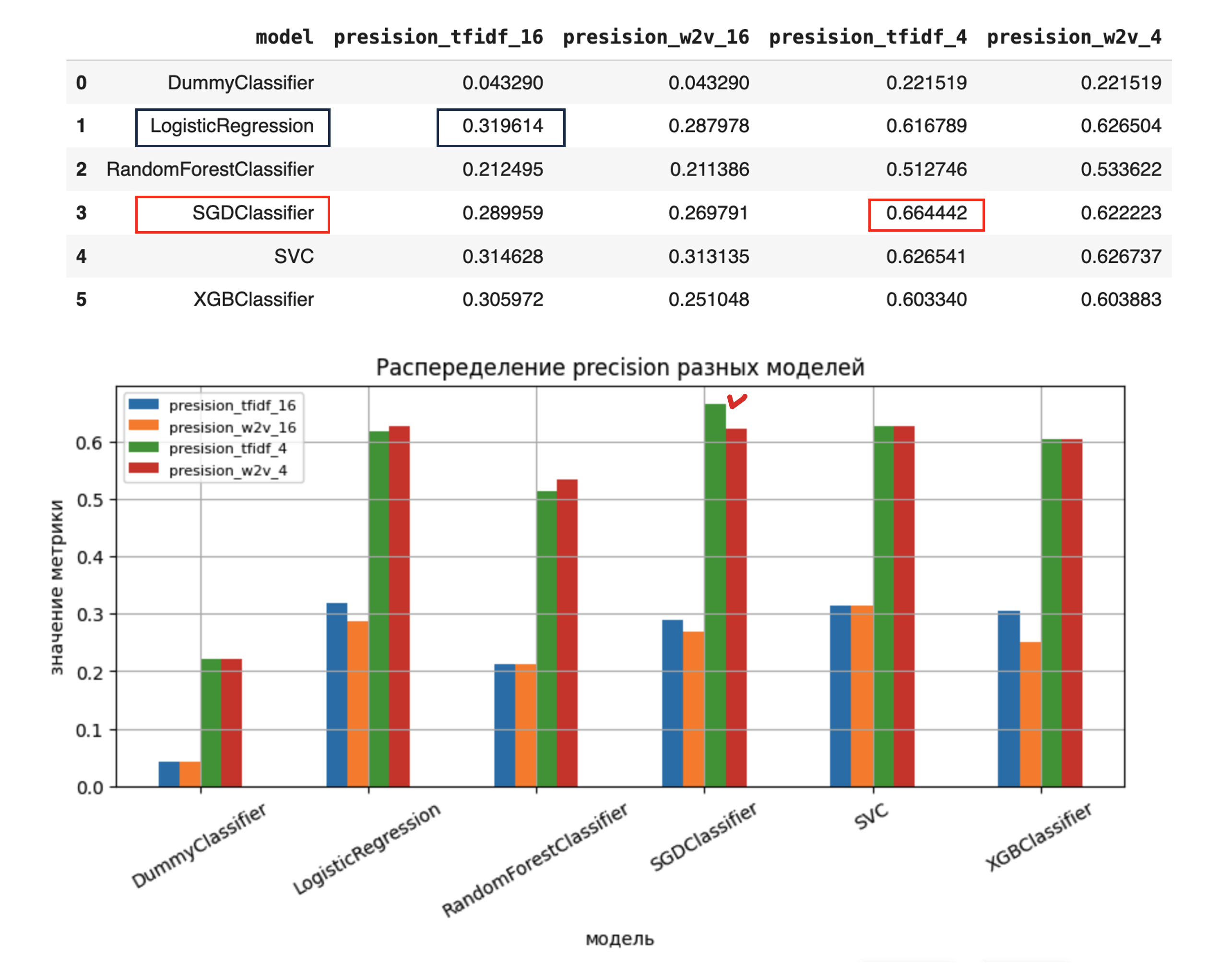
Total training took 0:23:08



Однако результаты меня не устроили (как по метрике и качеству, так как и по времени обучения), и я решила сфокусироваться на моделях классического ML.

Обучение с помощью классифческим ML я начала с DummyClassifier, который всегда предсказывает мажоритарный класс, его результаты указывают, на особенности данных и позволяют оценить насколько хороша финальная метрика.

Ниже я привожу сравнение для pressision для 5 различных моделей плюс DummyClassifier на 4 и 16 классах:



Лучший результат получился при обучении Tf-Idf векторов на 4 типа личности, модель **SGDClassifier – 66%.** Результаты DummyClassifier указывают на то, что моя точность в 66% довольная высока для данной задачи.

Достоинства **SGDClassifier:**

* Не требуется много ресурсов
* Высокая скорость работы

1. Итоговая модель

В качестве финальной была выбрана модель **SGDClassifier** (линейная модель со стохастическим градиентным спуском)

В качестве метода векторизации Tf- Idf ngram\_range=(1, 1)

С помощью GridSearch выполнен подбор оптимальных параметров

Разделение на тренировочную и валидационную выборки 80 к 20 (подобрано опытным путем)

Гиперпараметры подбирались с помощью GridSearchCV на 10 фолдах



Гипермараметры финальной модели:

**(alpha=0.001 (**оказывает значение на регуляризацию**), loss=‘modified\_huber**’, (сглаживает выбросы и основывается на расчете вероятностей)

**tol = 0.01,** (критерий остановки)

**max\_iter=10**, (максимальное количество проходов по обучающим данным)

**class\_weight='balanced’,** (cбалансированный режим, выполняет автоматическую корректировку весов)

**learning\_rate = 'optimal’** (выбирается с помощью эвристики)**)**

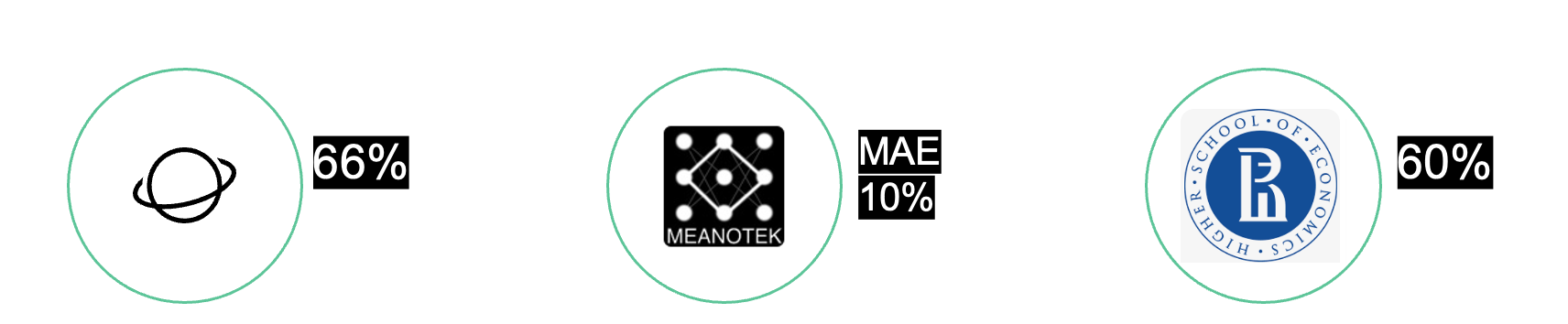
1. Сравнение с другими моделями

Мой результат в 66% позволяет поставить мое решение в один ряд с вышеупомянутыми решения от MeanoteK и ВШЭ.

Результат от ВШЭ:

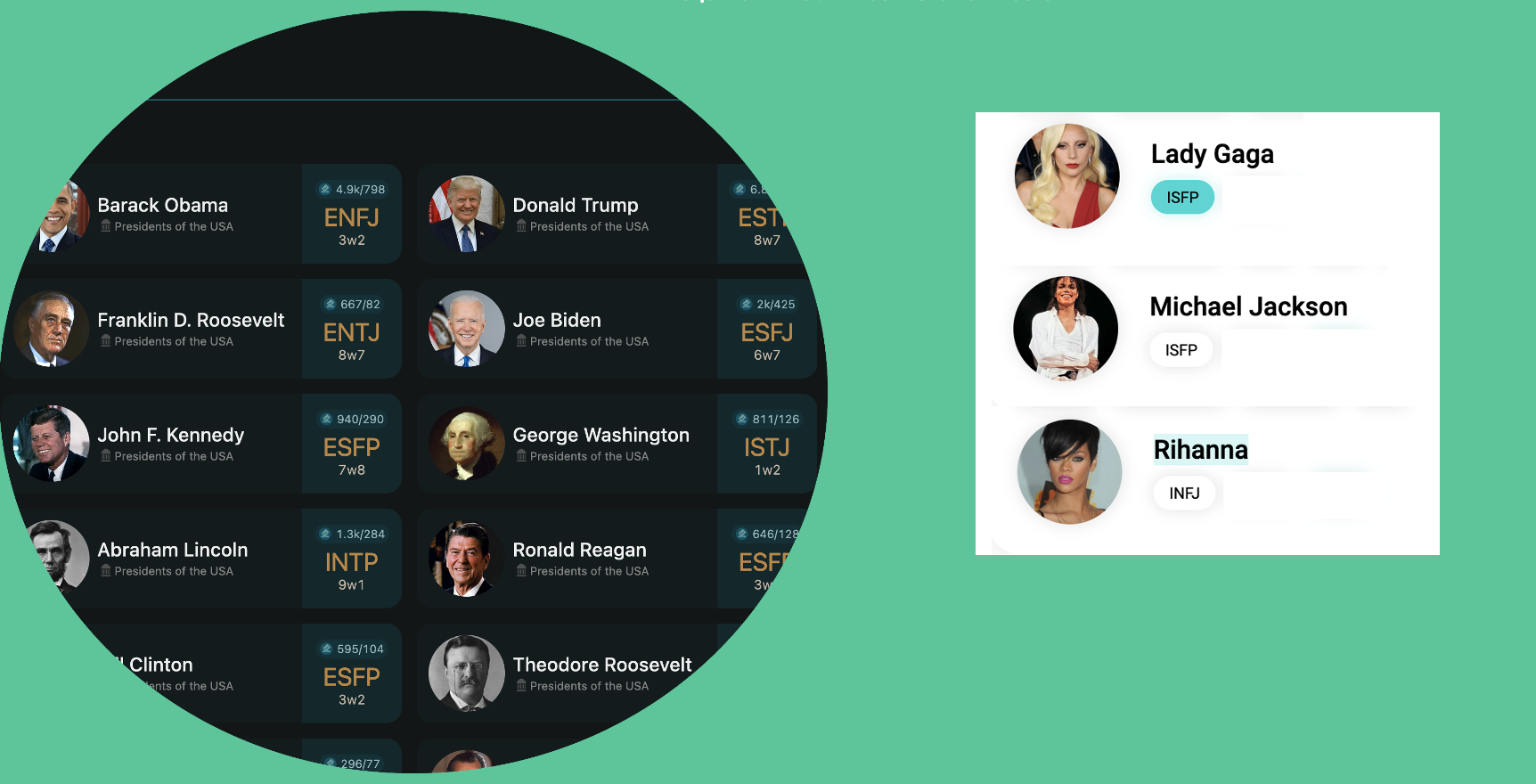
алгоритм делал верное заключение почти в 60% случаев, тогда как случайное угадывание обычно совпадает лишь в 50%. Превосходство в 10% кажется незначительным, однако на самом деле по точности предсказаний искусственный интеллект существенно опережает людей, если они судят по чертам лица незнакомого им человека.

MeanoteK использовали другую метрику MAE, которая составила около 10%.

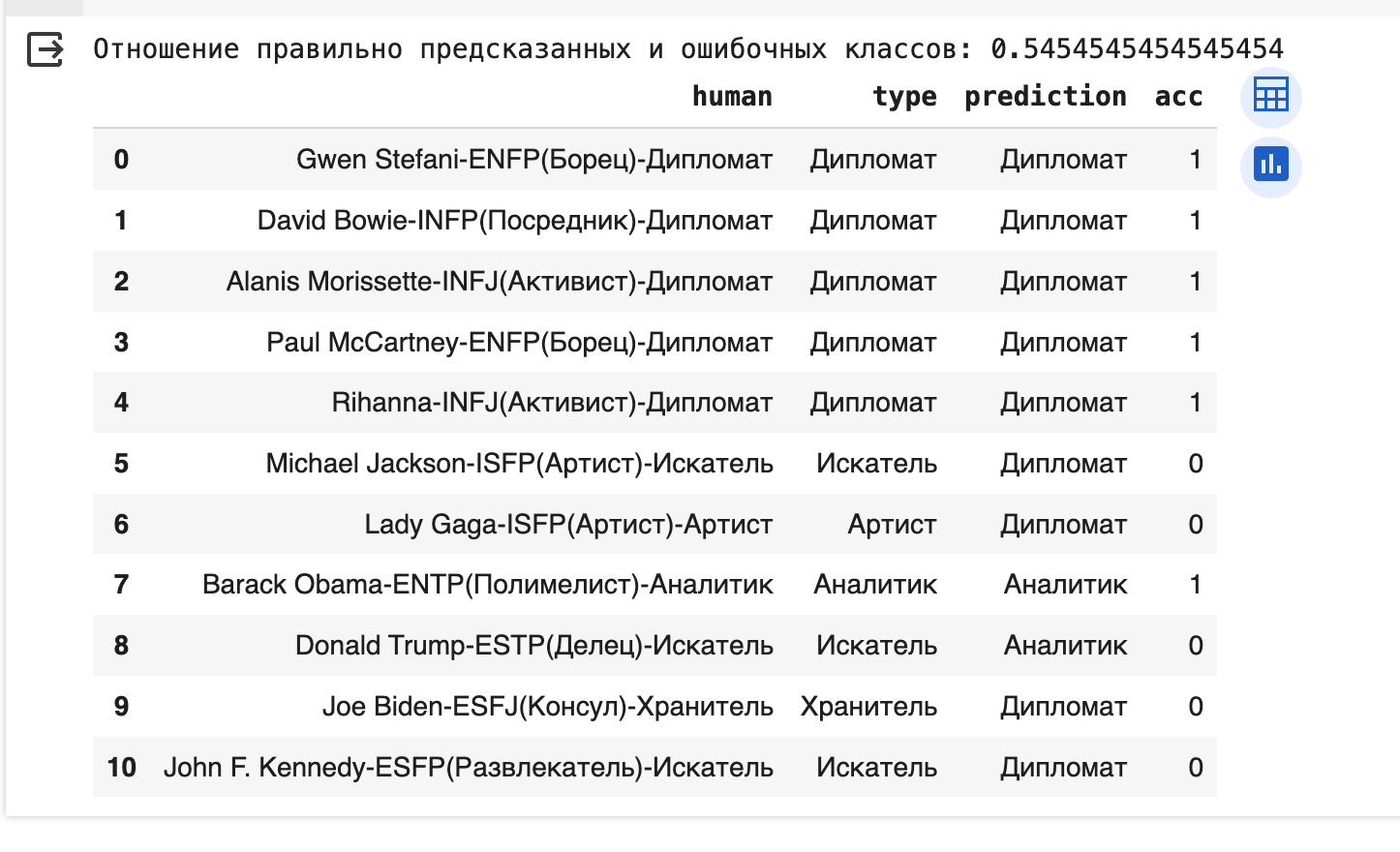


1. Проверка робастности модели на искусственно созданном датасете

Я собрала свой небольшой датасет из знаменитых людей и их текстов. Я взяла композиторов и певцов и их тексты песен, и президентов США и их высказывания из социальных сетей. В открытых сайтах опубликованы их типы личности по MBTI



В итоге я получила следующий результат:



Модель дает верное решение в 55% случаев, это лучше, чем при случайном угадывании.

Также следует принять во внимание, что достоверность данных можно поставить под сомнение, как те типы, которые опубликованы в открытых источниках, так и тексты, например за президентов тексты пишут их пресс-секретари и помощники. Эта таблица демонстрирует работу модели на новых данных не из базового датасета.

8.ВЫВОДЫ

Я считаю, что мое исследование показывает возможность совмещения психологии и методов искусственного интеллекта, так как есть востребованность на подобные сервисы. Результат в 66% довольно высокий с учетом качества базового датасета.

Хотелось бы в будущем адаптировать модель под русский язык.

Список используемых ссылок:

1. [Myers-Briggs® Overview](https://www.myersbriggs.org/my-mbti-personality-type/myers-briggs-overview/)
2. [Научно-образовательный портал IQ HSE](https://iq.hse.ru/news/366225961.html)
3. [Хабр MeanoTek Определение типа личности по тексту](https://habr.com/ru/companies/meanotek/articles/689360/)
4. [Датасет Kaggle](https://www.kaggle.com/datasets/datasnaek/mbti-type)
5. [Практикум по обработке естественного языка](https://www.youtube.com/watch?v=wBLITHbvAcc&list=PLBjWf0SyatqXlYjNRldqs2onXmNlhJ2xr&index=49)
6. [Метрики в задачах машинного обучения](https://habr.com/ru/companies/ods/articles/328372/)
7. [Онлайн тест личности](https://www.16personalities.com/ru/tipy-lichnosti)
8. [Personality-database](https://www.personality-database.com/), [Типы личности известных людей](https://boo.world/ru/database/famous-people-personality-types)