**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**



**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ Информатика, искусственный интеллект и системы управления

КАФЕДРА Системы обработки информации и управления

# Домашнее задание

# по дисциплине ****«Методы машинного обучения»****

**Тема: «Классификация тестовых сообщений»**

Выполнил: Нобатов И. - студент ИУ5-24М

Проверил: Гапанюк Ю.Г. - к.т.н, доц.

Москва, 2023 г.

1. **ВЫБОР ЗАДАЧИ**

Задачей является изучение способов классификации текстовых сообщений методами машинного обучения в контексте возможности автоматизации систем Service Desk.

1. **ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

Задачи классификации текста изучаются и реализовываются в программном виде уже в течение нескольких десятилетий. Интерес к применению классификаторов текста вырос в особенности с прорывами в области обработки естественного языка, которые произошли во второй половине 2010-ых годов. В большинстве систем классификации текста реализованы четыре основные фазы: извлечение признаков, сокращение размерности, выбор классификатора и оценка модели. [1]

Существует два основных подхода к выделению признаков из текста.

Первый подход основан на «взвешивании слов», то есть на преобразовании слова или иной обрабатываемой единице в такое скалярное значение, которое отражает частоту его использования в тексте. Наиболее активно используемыми техниками из этой группы являются «Weighted Words» и «TF-IDF», который присваивает слову значение пропорционально его использованию в отдельном тексте и обратно пропорционально использованию в остальных текстах пакета, что должно уменьшить влияние частоты использования слова в языке на выделение признаков.

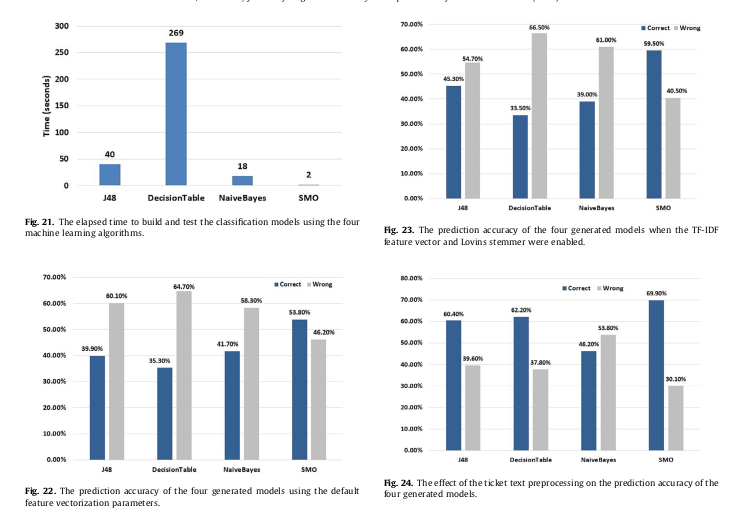
Второй подход основан на векторизации слов и способны лучше учитывать контекст использования слова. В методах векторизации(GloVe, FastText, Word2Vec) могут использоваться классические алгоритмы машинного обучения, а так же различные нейросетевые модели.

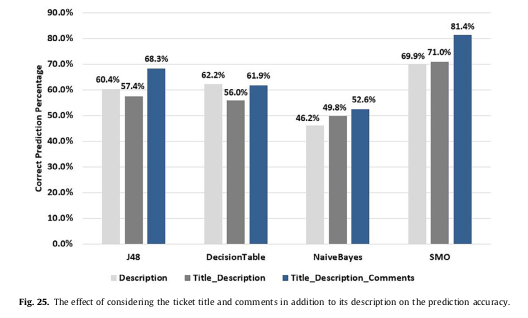
Для снижения размерности используются также как классические методы машинного обучения, так и новые методы основанные на нейросетях — автоэнкодерах. Используются такие методы как PCA(метод главных компонент), LDA (линейный дискриминантный анализ), автоэнкодеры, T-SNE и т.д.

Для задачи классификации также используются как методы классического машинного обучения, такие как алгоритм Роккио, логистическая регрессия, наивный байес, так и нейросети (рекуррентные, сверточные, автоэнкодеры).

Поскольку задача классификации заявок в Service Desk является довольно специфической, существует относительно не много исследований на эту тему. Часть из них посвящена изучению применимости классических методов машинного обучения. Рассмотриv их подробнее.

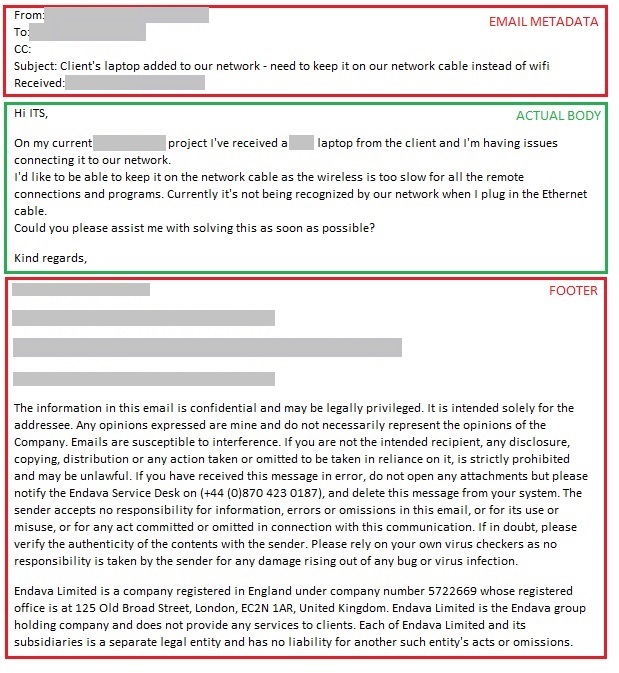
В статье «A machine learning based help desk system for IT service management»[2] за авторством Фераса Аль-Хавари и Халы Бахрам опубликованной в 2019 году изучается ряд моделей построенных с использованием ПО Weka, предназначенных для классификации поступающих заявок. На этапе выделения признаков во всех случаях используется модель векторного представления, а на этапе классификации каждая модель обучалась с использованием одного из классических алгоритмов машинного обучения, таких как SVM, Naïve Bayes, Rule-based и Decision Tree. Наилучшим образом себя показал алгоритм SVM-алгоритм на основе SMO. Изначально авторам удалось добиться точности в 53,8%, затем после применения предобработки и выделения в отдельные наборы заголовков и описаний 81,4%. Результаты тестирования моделей приведены на рисунке 1 и рисунке 2.

Рисунок 1 — Изначальные результаты обучения

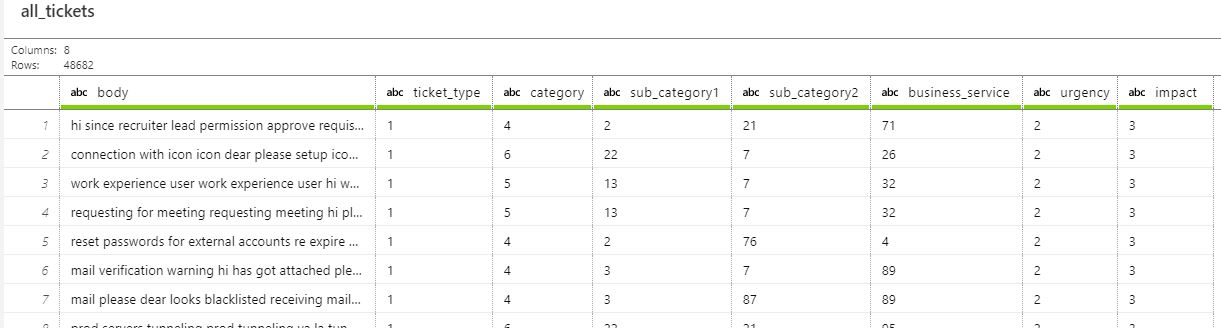
Рисунок 2 — Результаты после выделения названия и описания в отдельные наборы

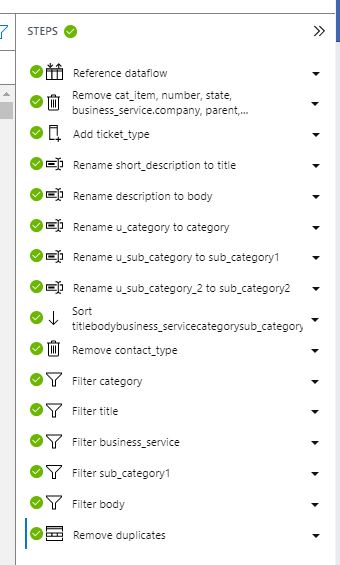
Можно заметить, что авторы не смогли добиться высокой точности на основном наборе данных. Поскольку даже результат точности в 81,4% был достигнут лишь на отдельно выделенном датасете с названиями запросов.

Рассмотрим исследование выполненное Microsoft в сотрудничестве с Envada.[3] Исследование также выполнялось с использованием специального ПО (Azure Machine Learning Studio, Microsoft Cognitive Toolkit, Azure Machine Learning service), его целью ставилось создание модели для автоматической классификации технических заявок для службы поддержки Endava. В качестве набора данных использовались внутренние данные Envada (50 тысяч классифицированных технических заявок с оригинальными сообщениями от пользователей и уже назначенными метками). Данные подвергались значительной предобработке и кодированию конфиденциальных данных, однако в рамках предобработки не была устранена несбалансированность набора данных, заключающаяся в том, что для некоторых классов заявок в датасете было слишком мало данных, что серьезно повлияло на результаты построения модели. То, как представлены текстовые сообщения, можно увидеть на рисунке 3.

Рисунок 3 — Представление текстовых сообщений

На рисунке 4 можно увидеть представление данных после обработки. На рисунке 5 продемонстрирована последовательность действий для обработки данных.

 Рисунок 4 — Представление данных после обработки

Рисунок 4 — Шаги при обработке

Несбалансированность данных продемонстрирована на рисунке 5.

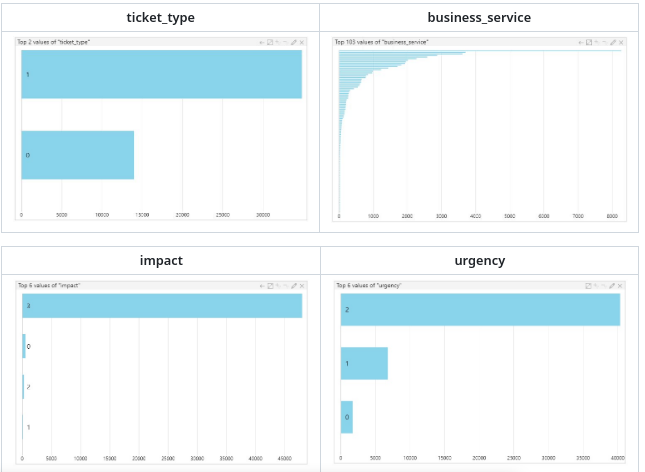


Рисунок 5 — Несбалансированность данных по некоторым столбцам

Для классификации были использованы 2 алгоритма классического машинного: SVM и наивный Байес. В обоих случаях результаты были довольно похожи, но для некоторых моделей наивный Байес показал значительно лучшие результаты. При обучении модели на наименее несбалансированном столбце (ticket\_type) были достигнуты наилучшие результаты, которые приведены на рисунке 6.

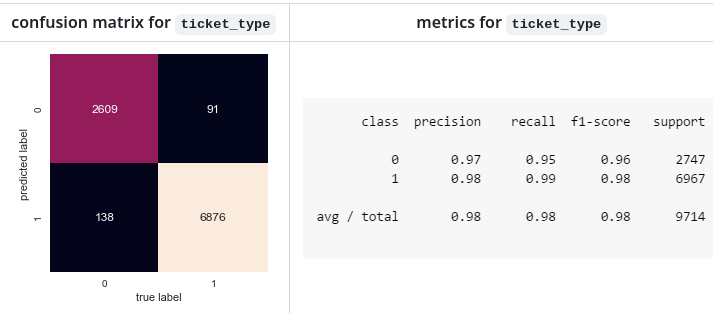
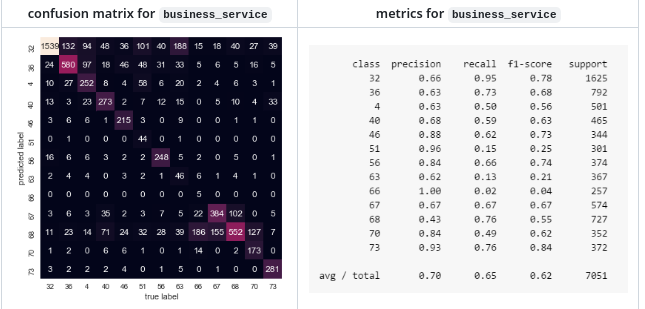


Рисунок 6 — Результаты обучения на первом этапе

Относительно приемлемые результаты были достигнуты для столбца business\_service. Для этого пришлось отбросить значительную часть значений с недостаточным количеством данных.(см. Рис. - 7)

Рисунок 7 — Результаты обучения на втором этапе

Для колонок category**,** impact **and** urgency не удалось достичь примелемых результатов из-за сильной несбалансированности. (см. Рис. - 8,9,10)

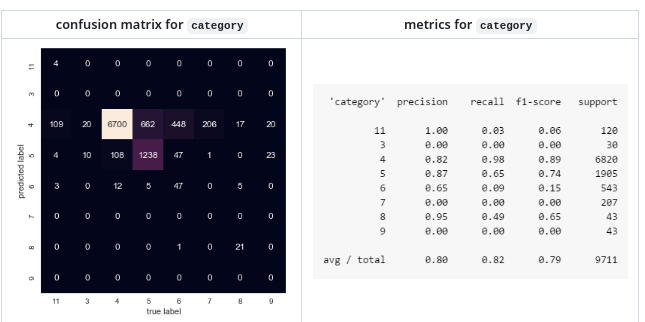
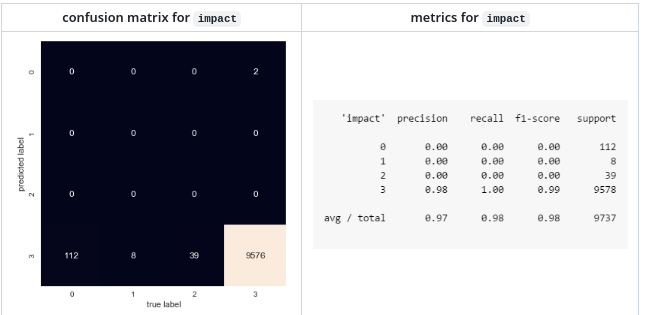
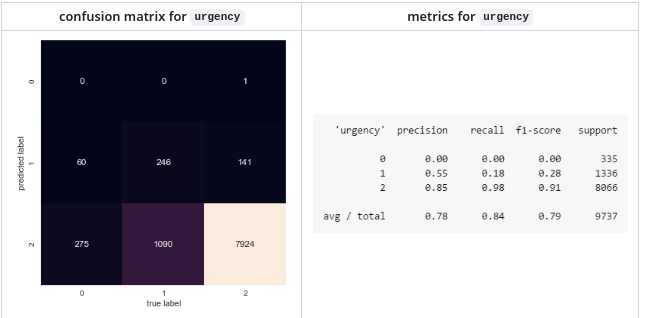


Рисунок 8 — Результаты обучения на третьем этапе

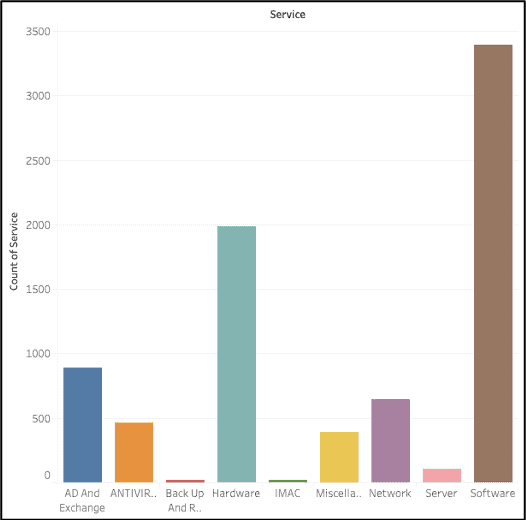
Рисунок 9 — Результаты обучения на четвертом этапе

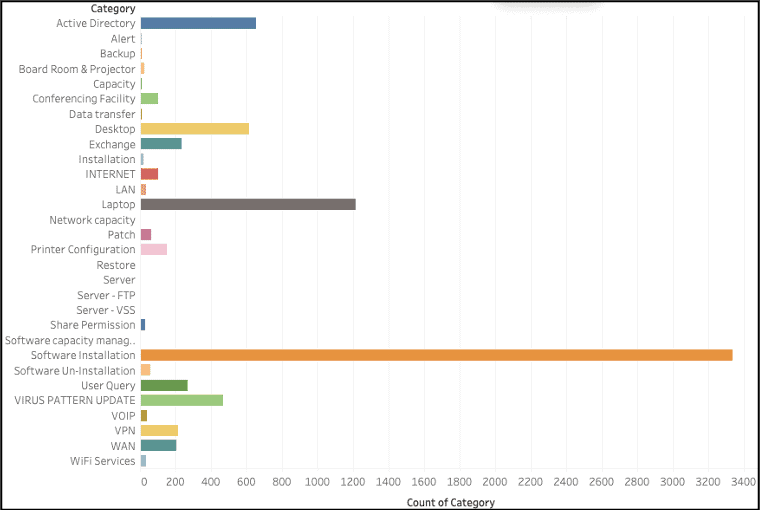
Рисунок 10 — Результаты обучения на пятом этапе

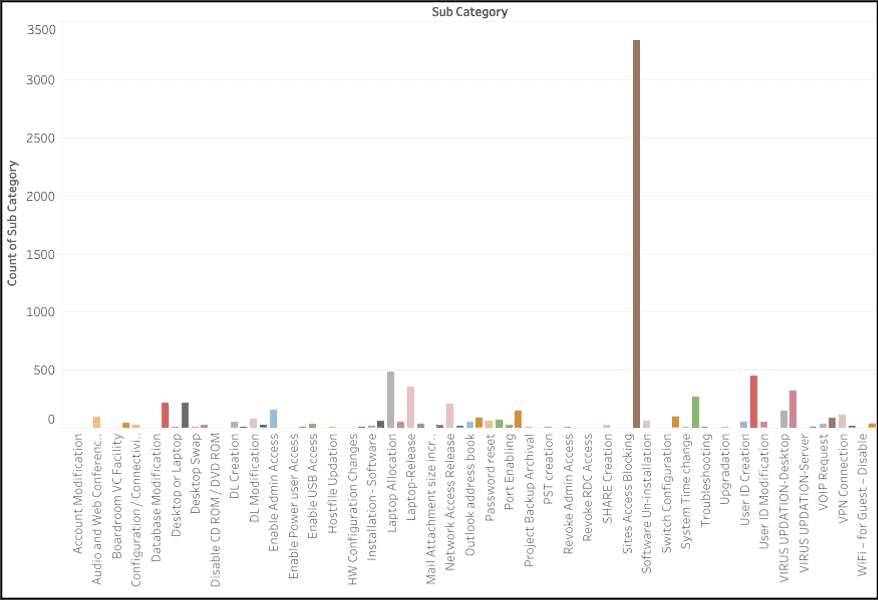
Датасет и исходный код, используемый в данном исследовании находится в открытом доступе. Разработчики в рамках исследования осуществили разработку веб-сервиса для предоставления доступа к модели. Исходники ПО также доступны в репозитории.

Рассмотрим еще одно исследование «IT Ticket Classification», представляющее интерес в контексте данной работы, опубликованное на сайте «analyticsinsight.net».[4]

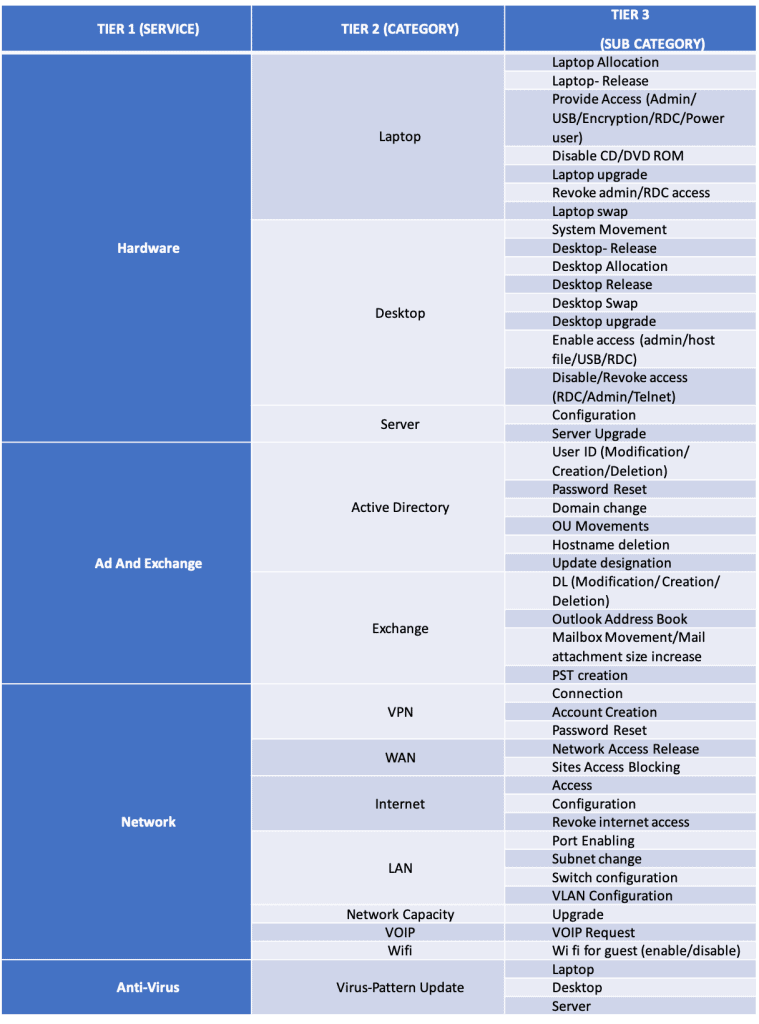
В данной работе ставилась цель построения модели, способной осуществлять классификацию на трех уровнях. Для обучения использовался соответствующий датасет. Распределение данных по категориям отображено на рисунках 11, 12, 13.

Рисунок 11 — Распределение по колонке «Service»

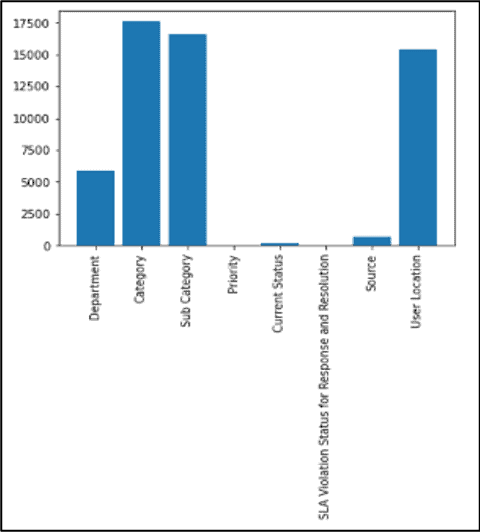
Рисунок 12 — Распределение по колонке «Category»

Рисунок 13 — Распределение по колонке «Sub Category»

Структура вложенности классов отображена на рисунке 14.

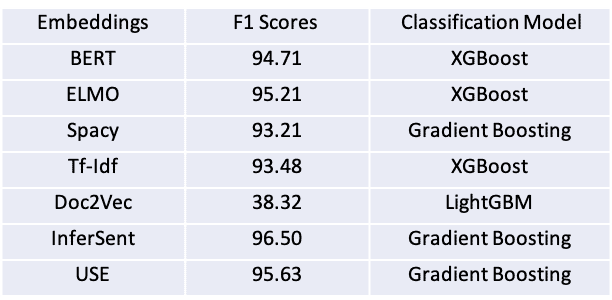
Рисунок 14 — Структура вложенности классов

Из исходных данных были выбраны только те столбцы, где по крайней мере 70% данных не были помечены как "NA". На выбранных столбцах были применены методы отбора признаков, чтобы определить, какие из них являются полезными. Поскольку независимые и зависимые переменные были категориальными, для понимания зависимости был использован тест хи-квадрат. Был использован алгоритм "SelectKBest" с методом хи-квадрат:

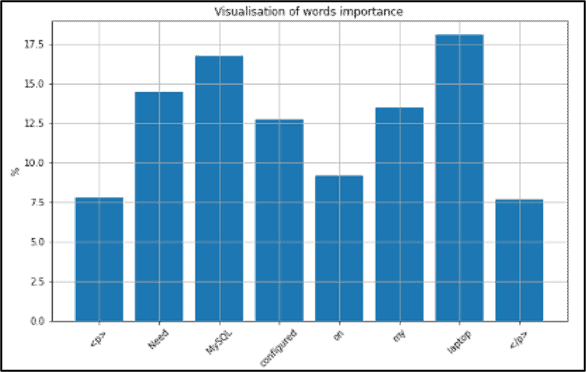
Рисунок 15 — Результаты

Ось Y представляет собой статистику хи-квадрат. Чем выше этот показатель, тем больше зависимость "Сервиса" от конкретной переменной. Только 'Отдел', 'Категория', 'Подкатегория' и 'Местоположение пользователя' имели значительный уровень зависимости с первым уровнем иерархии: 'Сервис'. Поскольку 'Категория' и 'Подкатегория' являются уровнями последующей иерархии, они не будут учитываться при построении модели.

На этапе выделения признаков авторы использовали различные методы. Результаты приведены ниже на рисунке 16.

Рисунок 16 — Результаты использования различных алгоритмов выделения признаков

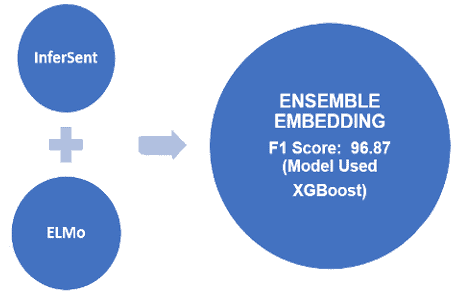
При использовании InferSent были достигнуты наилучшие результаты.(рис. - 17)

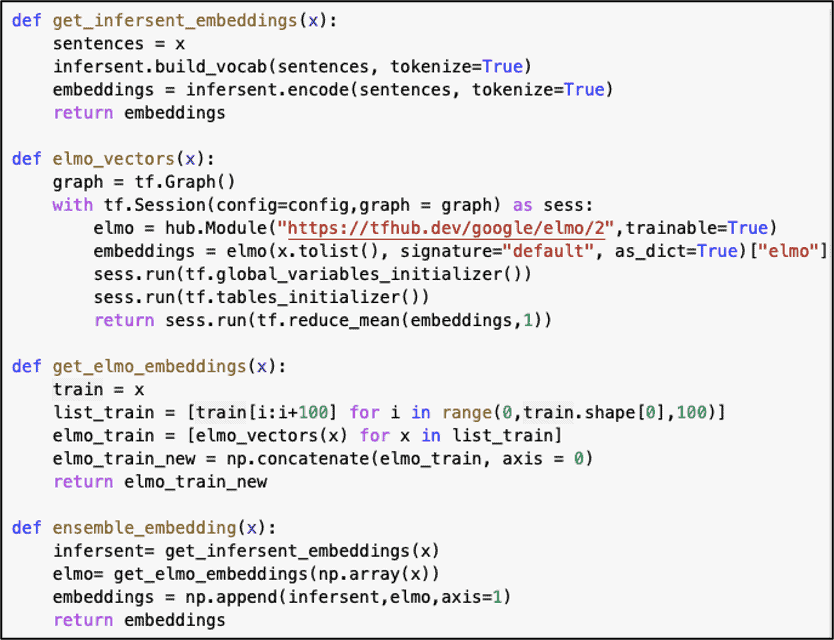
Рисунок 17 — Значимость слов при InferSent

Данные визуализации дают представление о том, насколько хорошо нейросеть InferSent удаётся распознавать ключевые слова в предложении. Таким образом, вышеуказанные визуализации показывают, что InferSent хорошо справляется с пониманием того, что «MySQL» и «Laptop» более важны, чем «Need», «on» или «my». Из визуализации можно также отметить еще один важный момент - что общие слова, используемые, например, «on», «my», имеют очень маленькую важность.

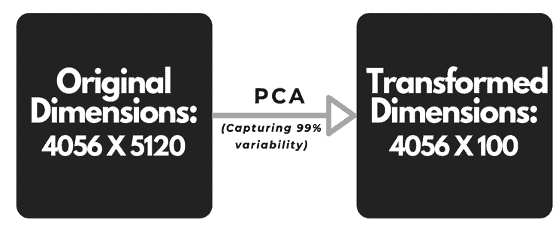
Поэтому метод встраивания становится довольно надежным, так как для текстовых данных не требуется традиционный этап предварительной обработки данных, такой как токенизация, лемматизация, удаление стоп-слов и т.д. Текстовые данные были переданы как есть, и метод работает весьма хорошо.

После этого были опробованы комбинации различных методов встраивания, чтобы увидеть, дадут ли ансамбли методов лучшие результаты. В итоге авторы остановились на следующем:

Рисунок 18 — Использованный вариант векторизации

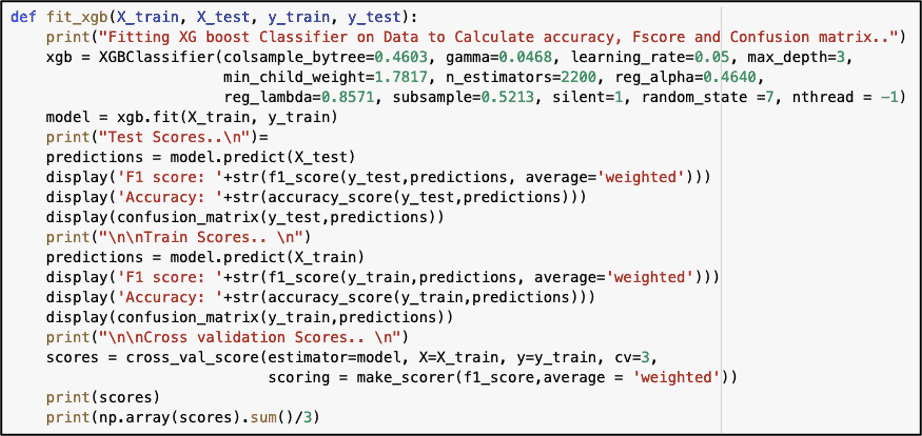
Рисунок 19 — Код реализации векторизации

Для уменьшения размерности, существенно выросших после векторизации данных, использовался метод главных компонент.

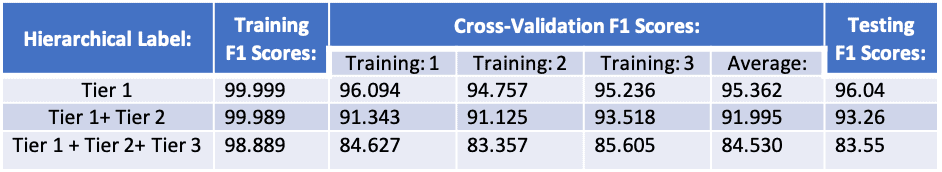
Рисунок 20 — Метод главных компонент для уменьшения размерности

Лучшая модель классификации была выбрана из нескольких алгоритмов машинного обучения на основе производительности на тестовых данных (не видимых для модели).

Модель XgBoost (экстремальный градиентный бустинг) лучше всего справилась с каждым уровнем, и код выглядел следующим образом:

Рисунок 21 — Экстремальный градиентный бустинг

Были достигнуты довольно высоки результаты.

Рисунок 22 — Показатели метрик модели

Стоит отметить, что в данном случае данные так же, как и в предыдущем рассмотренном исследовании, были сильно несбалансированы. Однако полученные результаты явно превосходят результаты предыдущей модели. По-видимому это может быть связано с нейросетевой векторизацией текста и возможно применением экстремального градиентного бустинга, который показывает себя гораздо лучше на несбалансированных данных.

Однако данная модель в качестве признака для классификации использует не только текстовое сообщение, но также данные о локации пользователя и данные о отделе компании, что в определенной степени усложняет сравнение модели с предыдущей.

Теперь рассмотрим статью «Ticket-BERT: Labeling Incident Management Tickets with Language Models» написанную группой сотредников Microsoft.[5]