деть, что самыми оптимальными вариантами являются такие биометрические методы идентификации пользователей как «По отпечатку пальцев» и «По венозному рисунку ладони». Они являются не слишком дорогими, но достаточно эффективными методами.

Литература

1. Руководящий документ Гостехкомиссии России «Защита от несанкционированного доступа к информации. Термины и определения». - М.: ГТК РФ, 1992. - 13 с.

2. Биометрическая идентификация [Электронный ресурс]/ российский интернет-портал и аналитическое агентство. URL: https://www.tadviser.ru/index.php/Статья:Биометрическая_идентификация_(мировой_рынок)#.

Д.С. Баканов

ПОСТРОЕНИЕ МОДЕЛИ ДЛЯ ПРЕДСКАЗАНИЯ ВРЕДОНОСНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

(Самарский университет)

Индустрия вредоносного программного обеспечения (ПО) продолжает расти. По оценкам лаборатории Касперского, лишь за один месяц 2020 года совершено свыше 10 млн. заражений [1]. И эта цифра продолжает расти.

Задача построения модели машинного обучения (МО), которая могла бы предсказать то, что ЭВМ в ближайшее время может быть заражена, была сформулирована компанией Microsoft на ресурсе Kaggle [2].

Данные представленные компанией Microsoft представляют таблицу (рисунок 1) с различными признаками (переменная в названии столбца таблицы). Исходом (признак, который надо предсказать) является переменная HasDetections, которая принимает значение 1 – данная машина заражена, 0 – иначе.

	MachineIdentifier	ProductName	EngineVersion	...	Wdft_IsGamer	Wdft_RegionIdentifier	HasDetections
0	0000028988387b115f69f31a3bf04f09	win8defender	1.1.15100.1	...	0.0	10.0	0
1	000007535c3f730efa9ea0b7ef1bd645	win8defender	1.1.14600.4	...	0.0	8.0	0
2	000007905a28d863f6d0d597892cd692	win8defender	1.1.15100.1	...	0.0	3.0	0
3	00000b11598a75ea8ba1beea8459149f	win8defender	1.1.15100.1	...	0.0	3.0	1
4	000014a5f00daa18e76b81417eeb99fc	win8defender	1.1.15100.1	...	0.0	1.0	1

Рисунок 9 – Внешний вид таблицы с данными

В данной таблице, как можно видеть из рисунка 1, каждый признак имеет разный тип данных. На рисунке 2 представлено распределение признаков по следующим типам значений:

– Категориальные признаки (Categorical Features) – признаки, которое имеют конечное количество строковых значений.

– Непрерывные числовые признаки (Numerical Feature) – признаки, которые имеют числовые значения из некоторого интервала.

– Бинарные числовые признаки (Binary Features) – признаки, которые имеют два числовых значения (например, 0 или 1) [3].

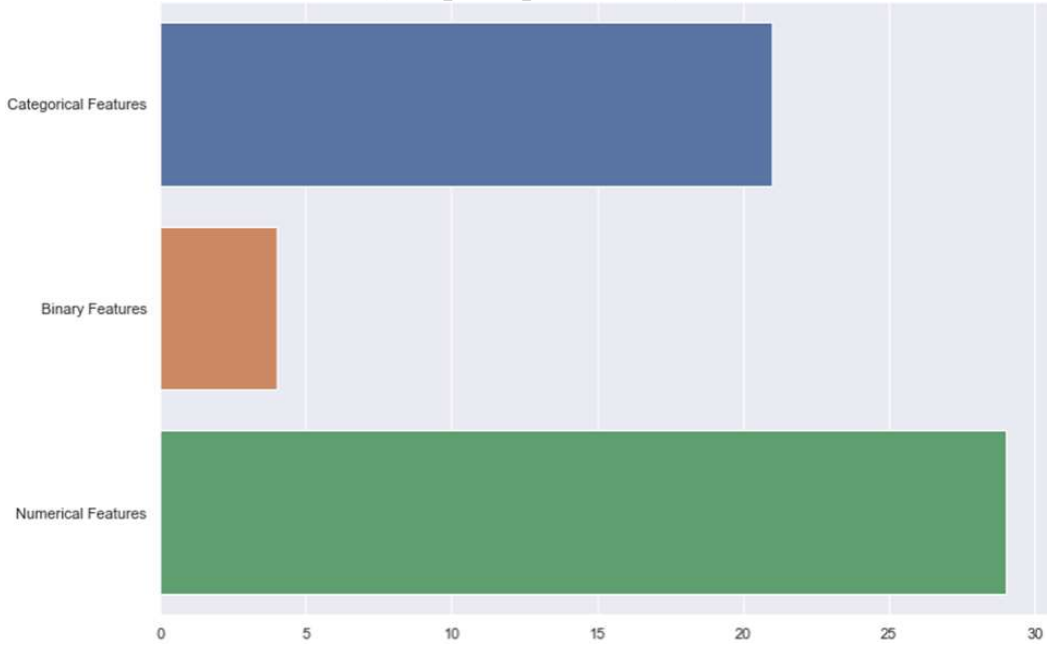

Рисунок 10 – Распределение признаков по типам значений

Как можно видеть, несмотря на то, что количество числовых признаков больше, категориальные данные тоже имеют существенное количество. Модель МО не сможет их обработать, поэтому можно провести факторизацию, т.е. сопоставить с каждой категорией какое-то число.

Для обучения модели также существенно распределение записей по классам исхода (т.е. записи, у которых HasDetections имеет значение 0 или 1). На рисунке 3 можно видеть, что дисбаланса среди записей нет, так как количество примеров почти одинаково.

Так как данных очень много (в изначальном наборе тренировочных данных насчитывается 83) были выявлены признаки с высокой долей пустых значений, а также с высокой долей часто встречающихся значений (свыше 0,9). Такие данные не играют высокой роли для обучения модели, поэтому их можно удалить из обучающего набора. Таким образом, удалось сократить количество признаков до 54.

Решаемая задача относится к задаче бинарной классификации, т.е. модель должна отнести каждую запись к одному из классов HasDetections 0 или 1. Основной сложностью является острая проблема масштабируемости данных из-за большого количества записей (8921483) в таблице и количества признаков. Но с другой стороны, такого количества данных не хватает, так как количество ЭВМ растет стремительно. Таким образом, нужно выбрать модель, которая хорошо обучается на малом объеме данных и в то же время нивелировала проблему масштабируемости данных.

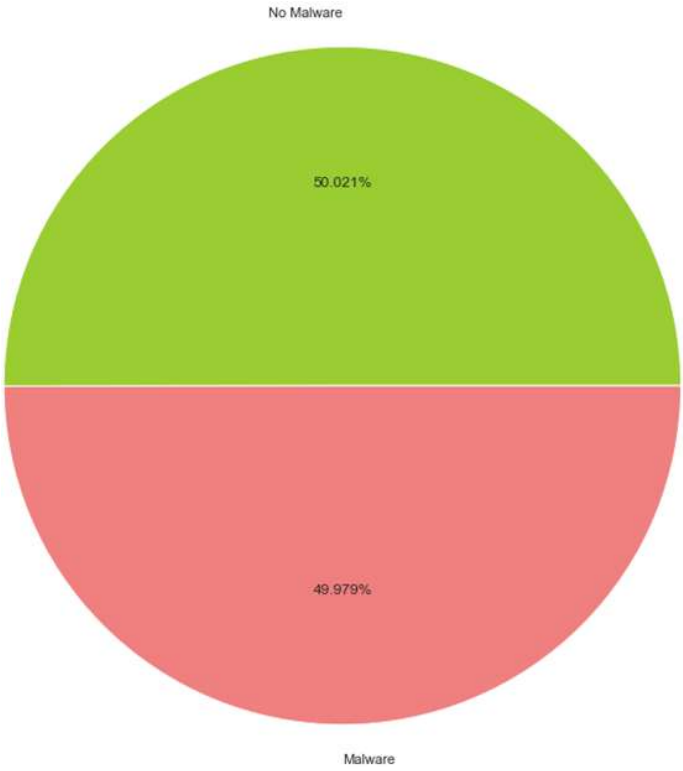

Рисунок 11 – Распределение записей по классам исхода

Проблему масштабируемости данных может решить модель решающих деревьев (Decision Tree). А в качестве алгоритма обучения можно использовать градиентный бустинг.

Градиентный бустинг – метод машинного обучения, который создает решающую модель прогнозирования в виде ансамбля слабых моделей прогнозирования, в нашем случае деревьев решений. Он строит модель поэтапно, позволяя оптимизировать произвольную дифференцируемую функцию потерь [4]. Порой градиентный бустинг выступает как отдельная модель МО. Данный алгоритм обучения хорошо подходит для обучения на малом объеме данных.

Для сравнения были использованы градиентный бустинг от компании Microsoftt LightGBM, алгоритм которого был описан в 2017 году [5], и CatBoost от компании Яндекс. Основным различием двух моделей являются алгоритм формирования обучающего набора для каждого дерева решений и возможность обработки категориальных признаков.

Перед обучением стоит определиться с метрикой качества модели. Так как решается задача бинарной классификации, то уместно использовать ROC-кривую, которая описывает зависимость чувствительности от специфичности. Специфичность (False Positive Rate – FPR) – процент (или доля) правильно классифицированных нулей. Чувствительность (True Positive Rate) – процент (или доля) правильно классифицированных единиц [3]. График представляется в виде «струны», которую слегка вывели из состояния покоя. А численное значение точности – это площадь под этой «струной».

На рисунках 4 и 5 приведены ROC-кривые для LightGBM и CatBoost соответственно.

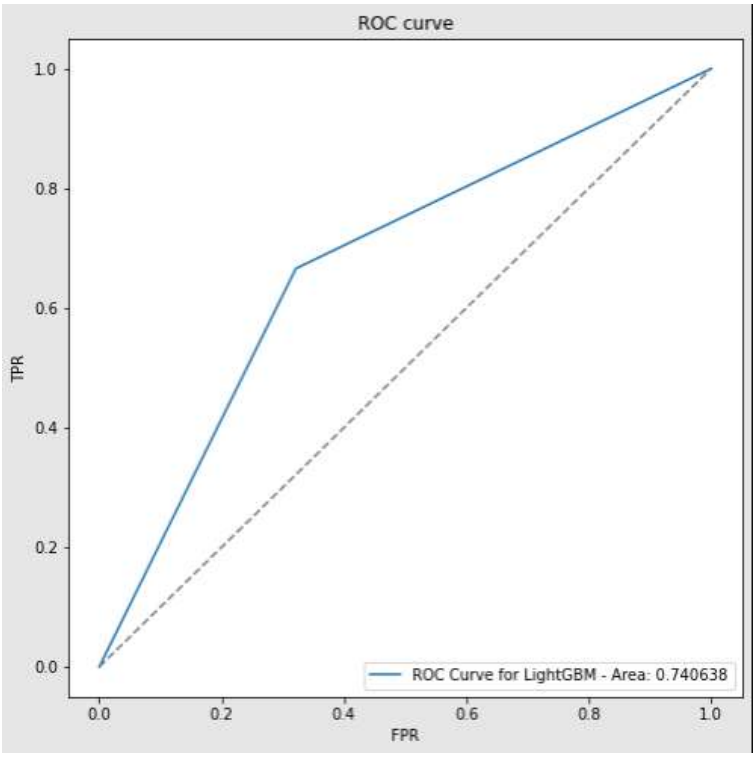

Рисунок 12 – ROC-кривая для LightGBM

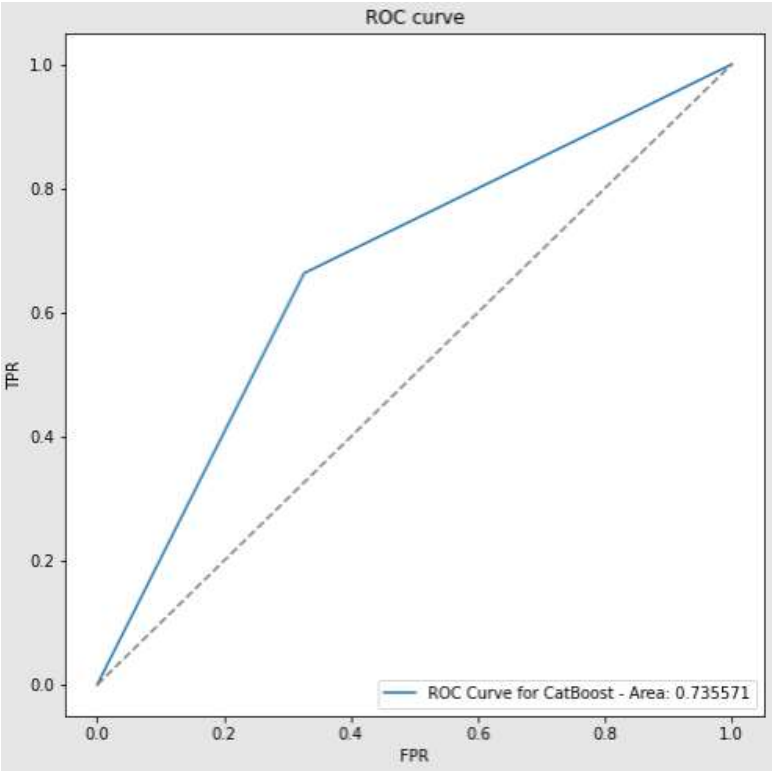

Рисунок 13 – ROC-кривая для CatBoost

Можно видеть, что обе модели показали хорошую точность (выше 0,7). LightGBM оказался более точным по сравнению с CatBoost (0,74 против 0,73). Однако скорость обучения CatBoost составила 35 минут, тогда как LightGBM – 44 минуты. Таким образом, обе модели показали достаточно хороший результат при обучении.

Таким образом, была произведена подготовка данных для обучения и создана модель МО, которая может предсказывать зараженность ЭВМ. Данную модель можно использовать в связке с антивирусом для быстрой оценки состояния системы в целом.

Литература

1 Интерактивная карта киберугроз [Электронный ресурс] // Kaspersky URL: https://cybermap.kaspersky.com/ru/stats (дата обращения: 20.10.2020).

2 Malware Prediction [Электронный ресурс] // Kaggle URL: https://www.kaggle.com/c/microsoft-malware-prediction (дата обращения: 20.10.2020).

3 Брюс, П. Практическая статистика для специалистов Data Science: Пер. с англ. / П.Брюс, Э. Брюс. – СПб.: БХВ-Петербург, 2020. – 304 с.: ил.

4 CatBoost [Электронный ресурс] // Университет ИТМО. URL: https://neerc.ifmo.ru/wiki/index.php?title=CatBoost#.D0.9E.D1.81.D0.BE.D0.B1.D0.B5.D0.BD.D0.BD.D0.BE.D1.81.D1.82.D0.B8_CatBoost (12.11.2020).

5 Friedman J.H. Greedy Function Approximation: A Gradient Boosting Machine // The Annals of Statistics. Vol. 29, No. 5 (Oct., 2001). C. 1189-1232

Д.Д. Габелия, Л.С. Зеленко

РАЗРАБОТКА ДИЗАЙН-СИСТЕМЫ ПРОГРАММНОГО КОМПЛЕКСА «КОНТРОЛЬ ОХРАНЫ ТРУДА»

(Самарский университет)

Основной частью процесса разработки программного обеспечения частью является этап проектирования будущей системы. Особое внимание на этом этапе следует уделить разработке пользовательского интерфейса, поскольку именно через него конечный потребитель сможет взаимодействовать с системой.

Разработкой полноценных интерфейсов обычно занимаются UI/UX-дизайнеры. В компании «СМС-Информационные технологии» (далее – «СМС-ИТ») проектированием прототипов экранных форм и последующим описанием требований к ним занимаются аналитики. Нередко аналитики задействованы на нескольких проектах одновременно, соответственно, в их задачи входит поддержание дизайн-проектов и систем требований к различным продуктам. Внутренние структуры соответствующих проектов и систем могут отличаться, что может вызывать затруднения в процессе работы и при адаптации к новому проекту.