MICROSOFT MALWARE PREDICTION COMPETITION

Калайчев Г.В.
Науковий керівник – канд. фіз.-мат. наук, доц. Сидоров М.В.
Харківський національний університет радіоелектроніки
(61166, Харків, пр. Науки, 14, каф. прикладної математики,
тел. (057) 702-14-36), e-mail: heorhii.kalaichev@nure.ua

The main goal of this work is to show the ways of preparation the amount of data, building a classification model on the huge dataset and evaluating resulting model on test data. Initial problem which was solved in this work was taken from Microsoft Malware Prediction Competition from Kaggle site. This task is an appropriate for our goal since training dataset contains different types of features for preprocessing and 9 million of rows.

Використання алгоритмів машинного навчання дозволяє розв'язувати досить складні задачі, які раніше не розв'язувалися взагалі, або це займало дуже багато часу та людських ресурсів. Наше дослідження проводилося з використанням CRISP-DM методології, яка складається з таких пунктів: розуміння бізнесу (мети роботи), розуміння даних, попередня обробка даних, розробка моделі, оцінка та розгортка програми.

1. Розуміння бізнесу та даних. Метою нашої роботи є передбачення імовірності комп'ютера Windows бути ураженим вірусами, яка заснована різних параметрах цього комп'ютера [1].

2. Попередня обробка даних. Набір даних налічує 81 стовпець, які мають бути використаними для передбачення ураження комп'ютера вірусом. Для розробки програми використовувалася мова програмування Python та такі бібліотеки: pandas (для роботи з CSV файлами та для розуміння даних); sklearn для обробки даних перед тренуванням моделі.

Стовпці мають різні типи даних, отже, попередня обробка – необхідний етап перед тренуванням моделі. Для стовпців, які мають категоріальний тип даних застосовувався one-hot encoding підхід.

Для обробки пропущених даних ми розрахували скільки пропущених даних знаходиться у кожному стовпці, потім ми видалили ті стовпці, у яких NaN значення складають більше ніж 60% від усіх значень стовпця. Решту NaN значень ми залишили, оскільки для навчання ми використовували моделі в основі яких лежать дерева, які не чутливі до пропусків у даних.

В результаті проведених операцій над даними ми отримали розріджену матрицю, яка налічує 1701 стовпець. Оскільки вхідні дані налічували 9 мільйонів рядків, зберігати усі дані разом у пам'яті не є можливим. Отже, для навчання моделі розріджена матриця була подана у вигляді блоків розміром 100 000 рядків, які завантажувались за вимогою.

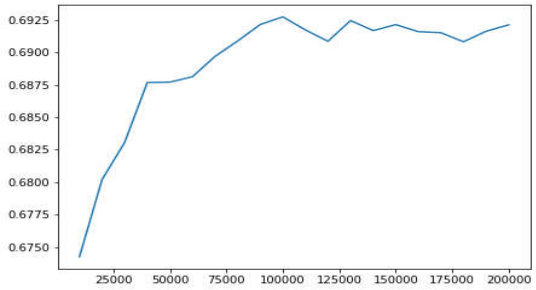
Також необхідним кроком було перемішати данні. Для цього ми використали tempfile пакет зі стандартної бібліотеки Python, який дозволяє зберігати інформацію на диску та має швидкий доступ до неї.

3. Моделювання та оцінка. Для розв'язання даної задачі нами був вибраний ансамблевий класифікатор XGBoost [2], як один з найсучасніших та ефективніших алгоритмів для роботи з великим об'ємом даних, які мають пропуски.

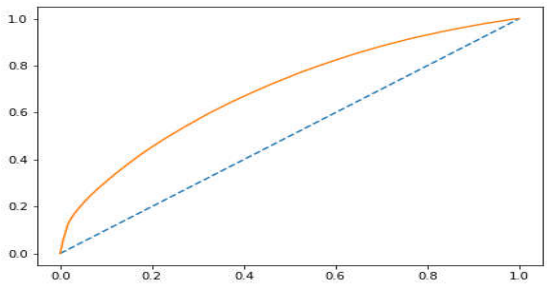
Основною проблемою під час моделювання були вибір моделі та підбір гіперпараметрів. За модель був обраний XGBoost, а для підбору гіперпараметрів був застосований досить популярний підхід Grid search with Cross validation.

Оскільки ми маємо великий набір даних, важливим є визначитися з кількістю рядків, необхідних для навчання стійкої моделі. Для цього ми навчали моделі з різною кількістю рядків та результати оцінок позначали на графіку.

На рис. 1 (а) ми бачимо, що 100 000 рядків достатньо для побудови стійкої моделі. Тестовий набір для оцінки моделі мав 1 мільйон рядків. Ми використовували Area Under Curve score для оцінки моделі, оскільки це базова метрика для задач класифікації. На рис. 1 (б) зображена ROC крива для найкращого класифікатора (моделі), який ми отримали.



(а) ROC AUC score для різного розміру даних



(б) ROC крива для кращої моделі

Рисунок 1 – ROC AUC scores та крива для кращої моделі

4. Результати. В результаті роботи була побудована модель, яка робить передбачення щодо можливості комп'ютера бути ураженим вірусом. При цьому найкращим результатом на тестовому наборі даних є 0.6927 ROC AUC score.

Список використаних джерел:

1. Microsoft Malware Prediction Competition, https://www.kaggle.com/c/microsoft-malwareprediction/data.

2. Tianqi Chen, Carlos Guestrin. XGBoost: A Scalable Tree Boosting System, 2016.