Status Report: Project 2, Malware Classification

Rajeswari Sivakumar • Vyom Srivastava •Nicholas Klepp

Overview

Progress

- Preprocessing
- Algorithms
  - Naive Bayes
  - RandomForest

Biggest issue

Computational limitations

Progress - Preprocessing

Accomplishment 1

- Using “Golden Features” recommended by Carl Liu https://www.youtube.com/watch?v=VLQTRILGz5Y&amp=&t=908s
- 1-4 grams Opcodes
- 1&2 grams Bytes
- Segment Counts

Progress - Model Training

Random Forest

- Trained Random Forest on:
  - 1 to 4 grams opcodes + segments
  - 1 & 2 grams opcodes
  - 1 & 2 grams bytecode
  - Unigram bytecodes
- Trees: 10/100, Depth: 4/10

  Max Accuracy : 27.6%

Naive Bayes

- Attempted to train Naive Bayes on just the byte code data (1 & 2 Grams)
- Never achieved scalable results.

Attention areas

Pyspark SQL DataFrames

- DataFrames - convenient when they work
- DataFrame → RDD
  + → DataFrame

  REALLY SLOW
- UDFs improve on this, but can be tricky to understand

Too Many Features

- 1, 2, 3, and 4 grams
  ==
  Explosive Feature Space
- Intelligent feature selection is key

Attention areas

3-4 Hour Training Times

- 2-3 runs in a day, max
- Sometimes less, if there are bugs in the code

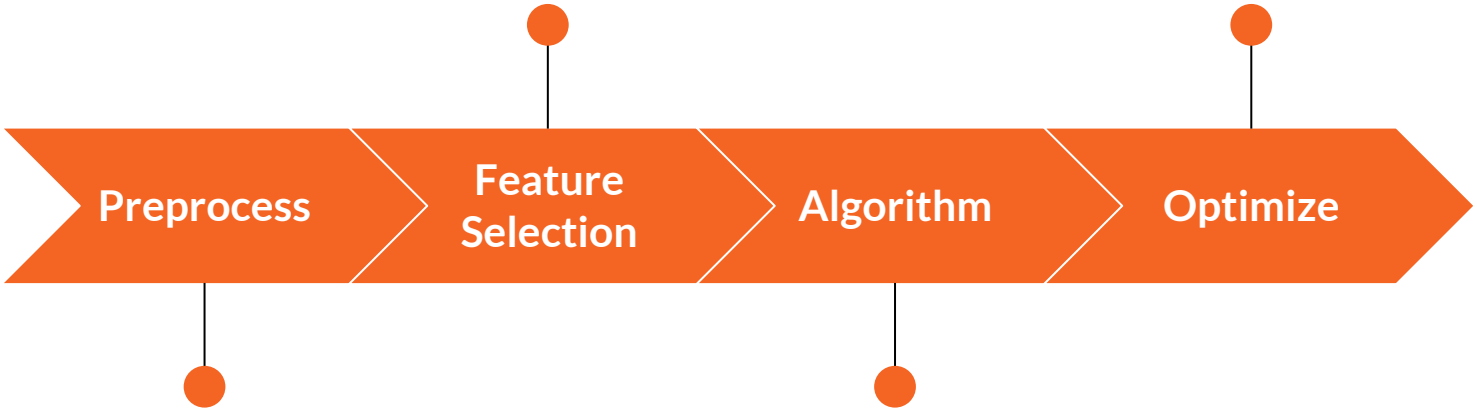
Panic

- When the time runs low, coding / software dev best practices get ignored

Work Flow

Chi Square Selector and limit vocab from Count Vectorizer

Improve memory management, parameter selection



Parse opcodes in asm files, 1-4 grams for byte files

RandomForest Classifier, Naive Bayes

Goals for improvement

1. Improve feature selection
2. Memory management
3. Debug lack of replicability