说明

王经 2018.01.30

dga_test: 测试每一个方法的正确率，并使用 classification_report 以及 confusion_matrix 对方法进行打分

dga_model: 导出每一种分类方法的模型

dga_model2: 对 dga_model 的 feature 为 2-gram 以及 234-gram 的模型导出方法进行修正，保存训练数据的词袋模型的词袋，并在之后预测中使用该词袋

dga_detect: 计算指定模型的给定数据的预测结果

dga_detect2: 保存了一个类，用于 dga_application 的应用

dga_application: 对给定数据计算多个模型的预测结果，并对结果进行对比，计算概率平均值

未解决问题：

1. 2-gram 模型的导入依然存在问题，即使对预测所得词袋模型进行了保存

可能原因: 我觉得可能是我变成上有问题，保存的是 CountVectorizor 这个类，感觉应该用类底下的函数 get_feature_name()函数，详见《Web 安全机器学习入门》P37

2. 最后平均值计算上，感觉应该使用加权平均数比较合理。我的建议是：可以使用 dga_test 中对每个方法的评分情况决定各种分类方法在加权平均值结果的占比问题。例如，如果 A 方法在 dga_test 中评分高，A 方法在平均值结果中占得权重就大；反之，占得权重就小。可以建立一个关系式，初步的想法为：

设方法$X_i(1 \leq i \leq n)$在dga_test中的评分结果为$Q_i(0 \leq Qi \leq 100)$,越接近100证明方法越好），方法X_i算得一个域名为DGA 的概率是$Pi(0 \leq Pi \leq 1)$,令$Q = Q_1 + Q_2 + ... + Q_n$,最后该域名为DGA的概率为为：

$$P = \sum_{i=1}^{n} \frac{Q_i}{Q} P_i$$