Proyecto 2: Predicción de infección por malware

Marchena, Sergio A.

Universidad del Valle de Guatemala
(mar16387@uvg.edu,gt)

***Resumen*—Este trabajo de investigación trata sobre el uso de algoritmos de Machine Learning y técnicas de manejo de datos para predecir infección en una colección de equipos con sistema operativo Windows por malware. La investigación consistió en un proceso de carga y exploración de datos, luego en un extenso trabajo de preprocesamiento de los mismos para poder llegar a desarrollar dos modelos de clasificación de machine learning (Decisión Tree Classifier y Gaussian Naive Bayes) para predecir sí estos equipos están infectados o no. Los resultados obtenidos son dos modelos con calificaciones de exactitud de 56%. Esto ayuda a Microsoft a utilizar controles preventivos antes de que una infección suceda.**

***Abstract*—This research work deals with the use of Machine Learning algorithms and data management techniques to predict infection in a collection of computers with a Windows operating system by malware. The research consisted of a data loading and exploration process, then an extensive preprocessing work to finally be able to develop two machine learning classification models (Decision Tree Classifier and Gaussian Naive Bayes) to predict whether these computers are infected or not. The results obtained are two models with an accuracy of 56%. This helps Microsoft use preventative controls before an infection happens.**

I. Introducción

este trabajo de investigación tiene cómo objetivo principal implementar modelos de machine learning que se basen en la telemetría de una computadora para calcular la probabilidad de ser infectada. Además tiene como objetivos secundarios comprender los datos de telemetría de Windows, comprender y explicar métricas de evaluación para la selección del mejor modelo y fomentar la investigación en ciencia de datos aplicados a ciberseguridad.

En este proyecto se realizaron dos modelos de clasificación de aprendizaje de máquina supervisado, los cuáles son Decision Tree Classifier, con una precisión de 56.61%, y Gaussian Naive Bayes, con una precisión de 56.31%. Ambos modelos tienen resultados muy similares, pero el primer modelo mencionado sería el mejor.

Cabe destacar que los datos proporcionados contaban con una colección de más de ocho millones de registros de computadoras, pero debido al límite de recursos para trabajar, se trabajó únicamente con un millón de datos seleccionados de forma al azar, para garantizar que se mantengan las proporciones de los datos.

II. Marco Teórico

Este artículo trata sobre temas sobre ciencia de datos aplicados a la ciberseguridad e infecciones de malware. Es por eso que vale la pena definir algunos términos clave para poder entender de mejor manera la investigación, el proceso, resultados y conclusiones obtenidos.

El malware [1] o "software malicioso", es un término general que describe cualquier programa o código malicioso que sea dañino para los sistemas.

Hostil, intrusivo e intencionalmente desagradable, el malware busca invadir, dañar o deshabilitar computadoras, sistemas informáticos, redes, tabletas y dispositivos móviles, a menudo tomando el control parcial de las operaciones de un dispositivo.

Los motivos detrás del malware varían. El malware puede tratar de ganar dinero con los usuarios, sabotear la capacidad para trabajar, hacer una declaración política o simplemente fanfarronear. El malware puede robar, cifrar o eliminar datos, alterar o secuestrar las funciones principales de la computadora y espiar la actividad de una computadora. sin conocimiento o permiso del propietario.

La telemetría es el registro y la transmisión automáticos de datos desde fuentes remotas o inaccesibles a un sistema de Tecnología de la Información en una ubicación diferente para su seguimiento y análisis.

[2] La telemetría de compatibilidad de Windows es un servicio en Windows Server 2019 que contiene datos técnicos sobre cómo funciona el dispositivo y su software relacionado. Periódicamente envía los datos a Microsoft para futuras mejoras del sistema y para mejorar la experiencia del usuario.

El modelo de árbol de decisión (Decisión Tree Classifier) [3] es un método de data mining de uso común para establecer sistemas de clasificación para desarrollar algoritmos de predicción para una variable objetivo. Este método clasifica una población en segmentos similares a ramas que construyen un árbol invertido con un nodo raíz, nodos internos y nodos hoja. El algoritmo no es paramétrico y puede manejar de manera eficiente conjuntos de datos grandes y complicados sin imponer una estructura paramétrica complicada.

Este modelo se puede ver de la siguiente manera:

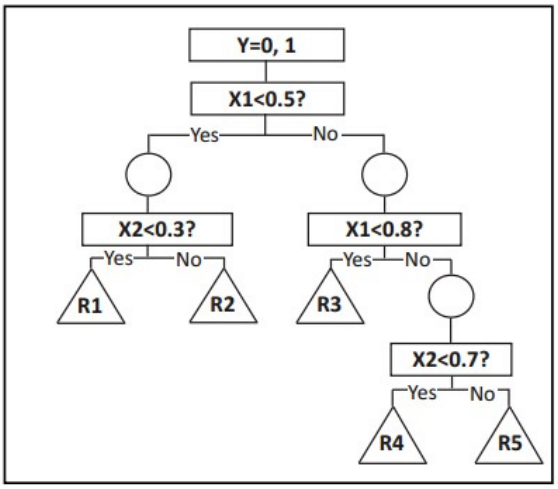

Fig. 1. Ejemplo de árbol de decisión basado en la variable objetivo binaria Y.

[3] El modelo de Gaussian Naive Bayes es una variante de Naive Bayes que sigue la distribución normal gaussiana y admite datos continuos.

Naive Bayes es un grupo de algoritmos de clasificación de aprendizaje automático supervisado basados en el teorema de Bayes. Es una técnica de clasificación simple, pero tiene una alta funcionalidad. Encuentran uso cuando la dimensionalidad de las entradas es alta. Los problemas de clasificación complejos también se pueden implementar utilizando Naive Bayes Classifier.

El modelo de Gaussian Naive Bayes asume que las características son independientes. Esto significa que se suponen matrices de covarianza específicas de clase, pero las matrices de covarianza son matrices diagonales. Esto se debe a la suposición de que las características son independientes.

El modelo se puede entender de la siguiente manera:

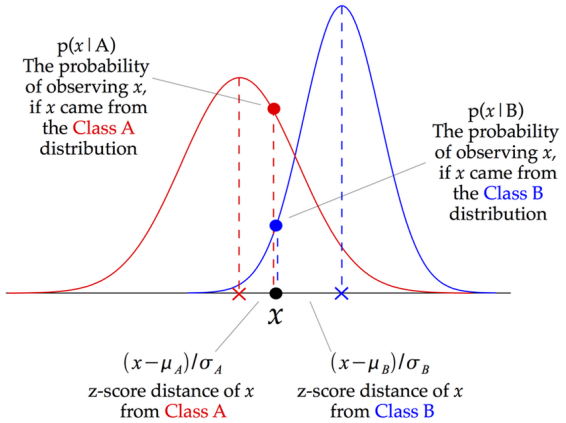

Fig. 2. Ejemplo de cómo funciona un clasificador Gaussian Naive Bayes.

III. Metodología

A. Análisis Exploratorio

El primer paso que se llevó a cabo en esta investigación fue un análisis exploratorio. En esta etapa se cargaron los datos originales y crudos. Después se exploran a simple vista las variables dentro de los datos, para poder entender los datos y ver si hay patrones o tener alguna idea inicial de los mismos.

B. Pre – procesamiento

Los datos luego de ser explorados y ya con una idea en mente del estado de los mismos deben de ser procesados para poder ser trabajados en un modelo. Los datos que eran de tipo categórico o texto se codificaron en numéricos. Luego usando la librería de Pandas Profiling, se generó un reporte con todas las descripciones de las variables y gráficas de las mismas. Este reporte también genera una matriz de correlaciones entre todas las variables. Este reporte se encuentra en el repositorio.

C. Selección de características

Con el reporte de los datos ya procesados generado en el paso anterior. Se llevó a cabo una prueba de correlación de Pearson de la variable objetivo: 'HasDetections' contra todas las demás. Con esta prueba se pudo seleccionar las características que más correlación tiene con dicha variable.

D. Implementación

El primer paso para la implementación de los modelos es separar los datos en datos de entrenamiento y datos de prueba. Se separaron los datos en una proporción de 70% datos de entrenamiento y 30% de datos de prueba.

Luego de esta separación, se procedió a implementar los modelos. Se implementó primero el Decisión Tree Classifier sobre los datos de entrenamiento y luego se hizo la predicción con los datos de prueba. Luego se implementó el Gaussian Naive Bayes con los datos de entrenamiento y por último se realizó la predicción sobre los datos de prueba.

E. Métricas de evaluación

Se obtuvo un reporte de métricas para cada modelo implementado así cómo su valor de exactitud en predicciones y una matriz de confusión. Las métricas indican valores de precisión, f1-score y support (promedio entre las dos anteriores). Esto nos da una idea para seleccionar o calificar el desempeño de cada modelo.

F. Evaluación del modelo

Por último, se evaluó cada modelo utilizando una curva de ROC. [4] Una curva ROC (curva característica operativa del receptor) es un gráfico que muestra el rendimiento de un modelo de clasificación en todos los umbrales de clasificación. Esta curva traza dos parámetros:

- Tasa de verdaderos positivos.

- Tasa de falsos positivos.

Además, se evaluaron los modelos por medio del método de validación cruzada de K-folds. La validación cruzada se usa principalmente para estimar la habilidad de un modelo de aprendizaje automático en datos no vistos. Es decir, usar una muestra limitada para estimar cómo se espera que funcione el modelo en general cuando se usa para hacer predicciones sobre datos que no se usaron durante el entrenamiento del modelo.

IV. Resultados

Según el análisis exploratorio y la prueba de correlación sobre la variable objetivo, las características seleccionadas para trabajar en los modelos implementados fueron:

EngineVersion	0.062320
AvSigVersion	0.069873
AVProductStatesIdentifier	0.117141
AVProductsInstalled	0.148466
Processor	0.073432
IsProtected	0.055844
Census_ProcessorCoreCount	0.053690
Census_TotalPhysicalRAM	0.058640
Census_OSArchitecture	0.072878
Census_IsVirtualDevice	0.052518
Census_IsAlwaysOnAlwaysConnectedCapable	0.062792
Wdft_IsGamer	0.055006

Fig. 3. Variables seleccionadas para trabajar en los modelos.

El modelo de Decision Tree Classifier obtuvo los siguientes resultados:

```
Accuracy:  56.61000000000001
Matriz de confusion:
 [[76727 73215]
 [56955 93103]]
              precision    recall  f1-score   support

           1       0.57      0.51      0.54    149942
           0       0.56      0.62      0.59    150058

    accuracy                           0.57    300000
   macro avg       0.57      0.57      0.56    300000
weighted avg       0.57      0.57      0.56    300000
```

Fig. 4. Resultados de Decision Tree Classifier

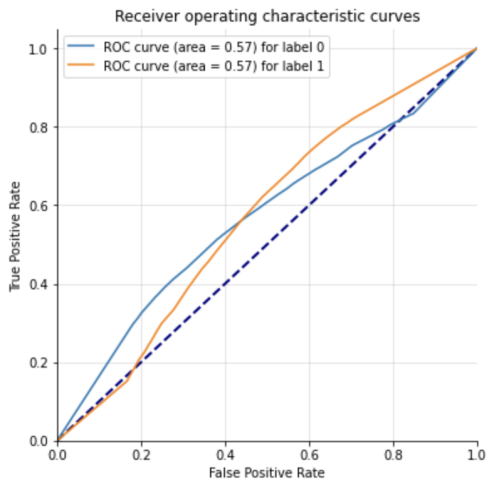

Fig. 5. Curva ROC de Decision Tree Classifier.

El modelo de Gaussian Naive Bayes obtuvo los siguientes resultados:

```
Accuracy:  56.382333333333335
Matriz de confusion:
 [[ 41865 108077]
 [ 22776 127282]]
              precision    recall  f1-score   support

           1       0.65      0.28      0.39    149942
           0       0.54      0.85      0.66    150058

    accuracy                           0.56    300000
   macro avg       0.59      0.56      0.53    300000
weighted avg       0.59      0.56      0.53    300000
```

Fig. 6. Resultados de Gaussian Naive Bayes.

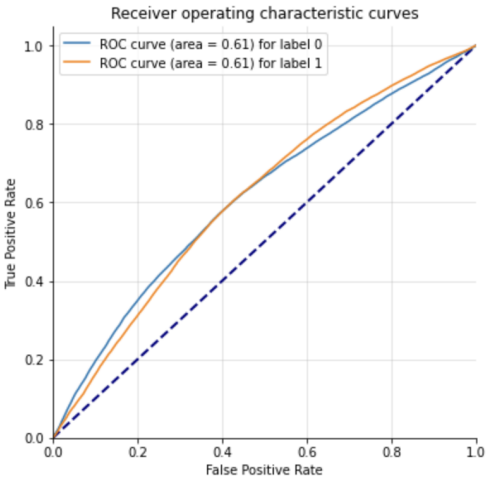

Fig. 7. Curva ROC de Gaussian Naive Bayes.

V. Conclusiones

- El modelo de Decision Tree Classifier es muy balanceado en la métrica de precisión, prediciendo con 57% y 56% las detecciones de malware en equipos Windows, pero tiene un valor de recall 51% y 62%, teniendo mejores resultados en no detecciones.

- El modelo de Gaussian Naive Bayes, en cambio, no es muy balanceado en la métrica de precisión, prediciendo con 65% y 54%, teniendo mejores resultados en detecciones de malware en equipos Windows. Este modelo tiene un valor de recall de 28% y 85%, teniendo mucho mejores resultados en no detecciones de malware en el equipo dado su telemetría.

- A pesar de tener mejor valor de accuracy el mejor modelo de predicción implementado con los datos proveídos, es el modelo de Gaussian Naive Bayes.

VI. Referencias

[1] Malwarebytes. (2022) What is malware? Malware definition. Recuperado de: https://www.malwarebytes.com/malware

[2] Microsoft Forums. (2021). Recuperado de: https://docs.microsoft.com/en-us/answers/questions/459823/how-can-i-turn-off-telemetry.html

[3] Song, Y. Y., & Lu, Y. (2015). Decision tree methods: applications for classification and prediction. Shanghai archives of psychiatry, 27(2), 130–135. https://doi.org/10.11919/j.issn.1002-0829.215044

[4] Google. (2020). Classification: ROC Curve and AUC. Recuperado de: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc