Universidad de Costa Rica

Escuela de Matemática y Ciencias Actuariales

Bitácora 2

Proyecto Herramientas de datos II

Integrantes

Gustavo Alberto Amador Fonseca C20459

Fabián Brenes Thomas C21380

Félix Madrigal Mora C24459

Marco Antonio Guardia Ortiz C23521

Laura Jimena Villacís Delgado C28386

Abril 2024

1 Objetivos

1. **Objetivo general:** Se plantea utilizar una base de datos de kaggle con observaciones de transacciones de tarjetas de crédito, con 31 distintas variables, 28 de las cuales son confidenciales por motivos de seguridad, otra que es el monto de la transacción y la variable a predecir: si es fraudulenta o no. Primeramente, se plantea realizar un análisis exploratorio de datos y posteriormente programar en phyton al menos tres modelos de machine learning: K vecinos más cercanos (KNN), Naive-Bayes, una regresión logística y algún otro modelo que pueda mejorar los anteriores en lograr predecir transacciones fraudulentas. Además, se quiere poder comparar los modelos de machine learning y lograr explicar por qué algún modelo puede ser mejor que el otro para esta tarea.

2. **Objetivos específicos:**

   - Realizar un análisis exploratorio de datos en Python, que abarque técnicas estadísticas y visuales para examinar tendencias y relaciones entre las variables, además de lidiar con observaciones sin datos y outliers, con el objetivo de facilitar la comprensión de la base de datos.
   - Programar los algoritmos de machine learning en python. A su vez, optimizar los procesos para agilizar el uso de los algoritmos.
   - Comparar el nivel de precisión de los modelos, mediante el porcentaje de exactitud para las observaciones de transacciones fraudulentas y para las observaciones de transacciones legítimas. Por último, desarrollar un breve análisis sobre por qué algún modelo funciona mejor que el otro.

2 Marco Teórico

El crecimiento del fraude en tarjetas de crédito ha sido constante desde la digitalización de las compras y del mundo financiero, lo que ha convertido este tema en uno de gran interés para el sector bancario. A lo largo de los años, se han desarrollado diversas técnicas para detectar transacciones fraudulentas, considerando variables como la fecha y ubicación de la transacción, la distancia del lugar de la transacción con la ciudad del dueño de la tarjeta, el tiempo transcurrido desde la última operación y la comparación del monto de la compra con el promedio de gastos del titular de la tarjeta, entre otros datos que, por razones de seguridad, suelen mantenerse confidenciales en los conjuntos de datos públicos. Además, es habitual que los estudios anteriores apliquen la técnica de Análisis de Componentes Principales (PCA) para reducir la dimensionalidad de los datos.

Comúnmente, para realizar este tipo de estudios, se han utilizado múltiples algoritmos de machine learning, entre ellos, la regresión logística, random forest, árboles de decisión, naive bayes y knn, así como métodos para la detección de anomalías. A continuación, se describen los métodos que se utilizarán durante esta investigación. Primero, el algoritmo de k-vecinos más cercanos (KNN) clasifica los datos basándose en su proximidad a observaciones previamente clasificadas con un conjunto de datos utilizado para entrenar al algoritmo. Este algoritmo emplea métricas de distancia, como la distancia euclidiana y la Manhattan, para realizar la clasificación de los datos. Se define previamente un parámetro entero, llamado k, que determina el número de vecinos a considerar a la hora de hacer la clasificación. La selección de este valor es crucial, pues valores distintos pueden conducir a un ajuste excesivo o insuficiente del modelo, como señala IBM (s.f.). Al introducir un nuevo set de datos para clasificar, se calcula la distancia mencionada y se seleccionan los k vecinos más próximos; una vez se obtienen estos, se clasifica la observación según la moda de sus vecinos, es decir, el grupo con más puntos alrededor de la observación. Además de clasificar, KNN también puede usarse como modelo de regresión para predecir variables continuas. Es importante destacar que: "El KNN es un modelo de aprendizaje perezoso, lo que significa que no pasa por una etapa de entrenamiento, sino que almacena un conjunto de datos de entrenamiento" (IBM, s.f.), por lo que una de las desventajas del algoritmo es el uso de más memoria con respecto a otros algoritmos de clasificación o regresión, lo cual puede significar mayores costos.

Este algoritmo puede ser utilizado para detectar transacciones fraudulentas bancarias. De hecho, Meera AlEmad lo hace en su tesis denominada: 'Credit Card Fraud Detection Using Machine

Learning'. En el trabajo se evidencia el uso de dos k distintos, el 3 y el 7, dando paso a resultados diferentes. Cuando se utilizó k igual a tres, se obtuvo una exactitud del 99,89% y con k igual a siete una de 99,88%. Lo cual sugiere que para este problema en concreto, un valor de 3 es lo más adecuado.

La regresión logística es un método estadístico, el cual es de los métodos de maching learning mayormente utilizados para la clasificación de dos clases; se utiliza para predecir la probabilidad de variables dependientes categóricas, donde las variables finales o resultados obtenidos van a ser dicotómicos, es decir solo van a tener dos opciones Verdadero/Falso. Este método consiste en tomar las variables aleatorias y conforme a distintas características que presenten se va analizar la probabilidad de que el evento ocurra o no, para esto se utiliza la función sigmoide que es propia de la regresión logística la cual viene dada por $P(y) = \frac{1}{1+e^{-y}}$ de forma que toma números de valor real y les asigna una probabilidad, si esta es mayor a 0,5 el resultado del evento estudiando es que si ocurre y de lo contrario no. A diferencia de la regresión lineal la salida que se va a obtener de la regresión logística es discreta.

En el caso que se está estudiando de la detección de fraudes en transacciones de tarjetas de crédito, la regresión logística puede ser muy útil ya que se puede utilizar en la clasificación de si la transacción es legítima o fraudulenta, por medio del análisis de las variables aleatorias y sus características, se les va asignar una probabilidad utilizando la función sigmoide y de esa manera se van a clasificar.

En cuanto al Naive Bayes, este algoritmo se fundamenta en el teorema de Bayes para calcular las probabilidades condicionales de los eventos. Aunque la suposición de independencia entre las características puede no ser realista, este algoritmo ha demostrado ser eficaz en tareas de clasificación; el algoritmo calcula la probabilidad de que un nuevo caso pertenezca a cada categoría utilizando el teorema de Bayes, para luego, seleccionar la categoría con la probabilidad más alta como la predicción para el nuevo caso.(Husejinovic, 2020). Utilizando datos históricos que incluyen transacciones crediticias tanto legítimas como fraudulentas, Naive Bayes predice si una nueva transacción es fraudulenta o no. Según Guseinov (2020), este método, junto con los árboles de decisión, ha logrado una tasa de éxito del 92,74% en predicciones.

Finalmente, el método de árboles de clasificación, como describe Sahin (2011), usa una estructura de árbol para dividir las observaciones en subgrupos más homogéneos mediante la creación

de un nodo raíz que se separa en nodos hijos. El proceso se repite hasta que no hay diferencia estadística significativa entre los grupos. Es crucial verificar que el árbol no sobreajuste las observaciones, lo que se puede corregir mediante la poda de nodos innecesarios, conocida como pruning. Ahora bien, en el artículo "Credit Card Fraud Detection Algorithm using Decision Trees-based Random Forest Classifier" (Madhubabu, 2021), se ofrece una vista previa de la implementación de árboles de decisión en Python sobre el fraude crediticio, demostrando la efectividad de este método.

En el ámbito del análisis de fraudes de tarjetas de crédito mediante técnicas de aprendizaje automático, Python se destaca por su eficacia, especialmente a través de la librería 'scikit-learn'. Esta biblioteca incluye módulos específicos para cada uno de los principales algoritmos utilizados en la detección de fraudes: KNN, regresión logística, Naive Bayes y árboles de decisión. Para comenzar con el método KNN, 'scikit-learn' ofrece el módulo 'sklearn.neighbors', donde la clase 'KNeighborsClassifier' es particularmente útil. Esta clase permite seleccionar el número de vecinos (k) y la métrica de distancia, facilitando así todo el proceso de clasificación. Es importante destacar que esta clase está diseñada específicamente para la clasificación; para aplicaciones de regresión, el mismo módulo proporciona clases adecuadas.

En cuanto a la regresión logística, 'scikit-learn' y 'statsmodels' son dos bibliotecas poderosas para su implementación. Mientras 'scikit-learn' es ampliamente usada para propósitos predictivos en general, 'statsmodels' ofrece herramientas más especializadas que pueden ser preferibles para análisis estadísticos detallados, incluyendo la regresión logística.

Para el uso de Naive Bayes, 'scikit-learn' también proporciona implementaciones eficientes que facilitan la aplicación de este algoritmo. La integración de Naive Bayes en Python a través de 'scikit-learn' simplifica la tarea de clasificar transacciones como legítimas o fraudulentas, basándose en la probabilidad calculada de acuerdo al teorema de Bayes.

Finalmente, para los árboles de decisión, 'scikit-learn' ofrece el módulo 'sklearn.tree', que incluye herramientas para crear, visualizar y optimizar árboles, con funcionalidades específicas para el pruning y mejoramiento del modelo.

3 Referencias bibliográficas

AlEmad, M. (2022) *Credit Card Fraud Detection Using Machine Learning.* https://repository.rit.edu/theses/11318/

Amat, J. (2020, noviembre) *¿Qué es KNN?.* https://www.ibm.com/mx-es/topics/knn

Duman, E., & Sahin, Y. (2011, marzo). Detecting Credit Card Fraud by Decision Trees and Support Vector Machines. *https://iaeng.org/.* Recuperado 3 de mayo de 2024, de https://www.iaeng.org/publication/IMECS2011/IMECS2011_pp442-447.pdf

Gonzalez, L. (2022, 23 septiembre). *Regresión logística Python.* https://cienciadedatos.net/documentos/py17-regresion-logistica-python

IBM. (s.f) *¿Qué es KNN?.* https://www.ibm.com/mx-es/topics/knn

Madhubabu, B. N. V., Vyshnavi, T., & Ashok, K. (2021). Credit Card Fraud Detection Algorithm using Decision Trees- based Random Forest Classifier. *Turkish Journal Of Computer And Mathematics Education.* Recuperado 3 de mayo de 2024, de https://turcomat.org/index.php/turkbilmat/article/download/11975/8760/21275

Scikit Learn(s.f.) *Nearest Neighbors* https://scikit-learn.org/stable/modules/neighbors.html

Scikit Learn(s.f.) *Scikit-learn, Machine Learning in Python* https://scikit-learn.org/stable/

Scikit Learn(s.f.) *sklearn.neighbors.KNeighborsClassifier* https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

Husejinovic, A. (2020, January 17). *Credit Card Fraud Detection Using Naive Bayesian and C4.5 Decision Tree Classifiers. Periodicals of Engineering and Natural Sciences, 8(1), 1-5.*

https://ssrn.com/abstract=3521283

Zhang, H. (2004). *La optimidad del ingenuo Bayes. AA, 1(2), 3.* https://cdn.aaai.org/FLAIRS/2004/Flairs04-097.pdf

Fernández Jauregui, A. (2023, 19 de agosto). *Clasificación de texto con Naive Bayes en Python.* https://anderfernandez.com/blog/naive-bayes-en-python/