**Proyecto Final Minería de Datos, 2022-2**

**Autor: Wilmer Arley Rodríguez Ropero**

1. **Introducción**

La minería de datos, a pesar de ser un campo de estudio relativamente nuevo, ha ofrecido una gran ayuda en la era moderna puesto que su aplicación, y con ayuda de otros campos, ha permitido **construir conocimiento** a partir de la gran cantidad de datos que se manejan hoy en día, esto ha llevado a implementaciones en diferentes áreas del conocimiento. Al final lo que se quiere lograr con esto, es hacer comprender que la minería de datos no simplemente está dirigida para el ámbito académico, si no que su uso en la vida cotidiana es muy conveniente, es por eso por lo que el ejemplo que trabajaremos está enfocado a determinar si una persona se suscribirá o no en un crédito de un banco. El fin de este documento es dar un breve abrebocas de lo que concierne al campo de estudio de la minería de datos, donde tocaremos diferentes técnicas y algoritmos para la obtención de dicho **conocimiento** que esta oculto en los datos en bruto.

1. **Definición del problema y algoritmo**

**2.1 Definición del problema**

Los bancos siempre se han preocupado por mantener sus ingresos estables, es por esto que se realizan campañas para hacer que la gente se suscriba a los créditos que estos ofrecen. Sin embargo, esto trae consigo un problema grave y es determinar si la persona acepta o no el crédito, y aunque a primera impresión no parece ser tan problemática esta situación, se tiene que entender que los bancos son instituciones que no pueden tolerar costos que a la larga no le están retornando ningún beneficio.

Con lo anterior en mente vamos a tratar los datos que nos han proporcionado para construir un modelo que sea capaz de predecir y reconocer que personas están dispuestas a recibir un crédito del banco, y así de paso hacer que el costo de oportunidad del banco se maximice y sea rentable para estos.

**2.2 Algoritmos**

En minería de datos existen varios modelos para la clasificación de los datos, es por esto por lo que implementaremos 3 diferentes técnicas de clasificación supervisada, las cuales detallaremos más adelante, además trabajaremos con un método clasificación no supervisada para poder ver que patrones podemos encontrar entre los datos.

Entrando en detalle con la clasificación supervisada vamos a dar una breve explicación de los modelos que vamos a usar, y veremos muy por encima cómo funciona cada uno de ellos, los modelos que usaremos son los siguientes:

* **Naive Bayes:** Es un modelo muy popular en el machine learning, su base teórica principal es el teorema de Bayes donde se trabaja con probabilidades de fondo para la clasificación de los datos. Además de esto trabajaremos 4 variantes de este modelo que son las siguientes:
  + GaussianNB
  + MultinomialNB
  + BernoulliNB
  + CategoricalNB

A pesar de que este modelo sea muy sólido puesto que presenta buenos resultados, tiene un problema que se tiene que considerar dependiendo de los datos que se estén manejando, y es que, de fondo para él, todos los atributos los considera como sucesos independientes, cuando en la vida real esto rara vez pasa.

* **Random Forest:** Este modelo, al igual que Naive Bayes, es muy usado en el machine learning, un bosque está formado por un conjunto de árboles de decisión que se entrenan con muestras aleatorias individuales para cada uno, lo que es mejor conocido como bootstrapping. El Random Forest es uno de los mejores modelos cuando se habla del control del bias-varianza ya que al trabajar de fondo con varios árboles de decisión esto impide que el modelo tenga overfitting y a la vez que tenga buenos resultados con otros datos externos a los de entrenamiento y prueba.
* **Support Vector Machine:** Es uno de los modelos con más base teórica que trabajaremos, toca temas complejos como lo son los hiperplanos, vectores de soporte, optimización, transformaciones, entre otros. Este modelo en un principio se creó para casos de pocos atributos, pero gracias al avance de la tecnología este algoritmo ha ganado popularidad por su rigurosidad a la hora de clasificar, sin embargo, dentro de los modelos, que se usaran, es uno de los que exige más capacidad de cómputo.

Ahora que hemos hablado sobre los métodos de clasificación supervisada, pasaremos a hablar sobre el método no supervisado, al igual que en los anteriores métodos tocaremos un poco la teoría que hay detrás del modelo. El método que trabajaremos será el siguiente:

* **Density Based Spatial Clustering of Applications (DBSCAN):** Como bien indica el nombre este modelo está basado en la densidad de los datos, es decir que tan aglomerados están. Sin embargo, cuando hablamos de densidad se tiene que considerar dos factores importantes, en primera medida tenemos que ver cuánto es la cantidad mínima de puntos para que un área sea considerada densa y en segunda instancia se tiene que observar cual es la distancia máxima tolerable entre estos puntos. Estos valores juegan un papel muy importante en el modelo ya que a partir de estos se crean los clústeres, por medio del siguiente algoritmo:
  + Calcula la matriz de distancia entre los diferentes puntos. Por lo general se usa la distancia Euclidiana.
  + Teniendo en cuenta los valores de distancia máxima entre puntos y cantidad mínima necesaria de puntos para que sea un área de densidad, el algoritmo clasifica los puntos entre las categorías core points, achievable points y noise.
  + Asigna los core points cercanos a un clúster especifico.

Como se puede observar es un algoritmo bastante sencillo y que es fácil de entender, algo interesante de este modelo es que en un principio no se le da la cantidad de clústeres a trabajar si no que, a partir de la distancia y puntos necesarios, este crea a consideración el clúster.

1. **Evaluación Experimental**

**3.1 Datos**

El dataset que manejamos provienen de distintas campañas realizadas por el banco institucional portugués, a partir de estas campañas se recolectaron un total de 45211 atributos con 17 atributos que son los siguientes:

* Age: Edad de la persona (Numérica).
* job: Ocupación (Categórica).
* marital: Estado marital (Categórica).
* education: Educación de la persona (Categórica).
* default: Si la persona tiene crédito en mora (Categórica).
* balance: Saldo promedio anual (Numérica).
* housing: Si la persona posee préstamo de vivienda (Categórica).
* loan: Si la persona posee préstamo personal (Categórica).
* contact: Modo de comunicación con la persona (Categórica).
* day: Último día en que fue contactada la persona (Numérica).
* month: Último mes en que fue contactada la persona (Categórica).
* duration: Tiempo que tomo la última comunicación con la persona, en segundos (Numérica).
* campaign: Cantidad de comunicaciones realizadas a la persona durante una campaña, incluye la última vez que fue contactado (Numérica).
* pdays: Número de días que pasaron después de que el cliente haya sido contactado para campañas realizadas anteriormente (Numérica).
* previous: Cantidad de llamadas realizadas antes de la campaña actual (Numérica).
* Poutcome: Resultado de campañas realizadas anteriormente (Categórica)|
* Y: Esta o no suscrito al crédito (Categórica)

En un principio este dataset no posee ningún dato nulo, sin embargo, por cuestiones académicas se hizo una inserción aleatoria de datos faltantes que correspondía al 10% del total, en este paso no se le inserto datos nulos a la clase. El objetivo de haber hecho lo anterior es poder mostrar cómo se realiza un preprocesamiento básico a los datos, este es de los pasos que más toman tiempo en la minería de datos, después de todo, así como los modelos reciben y dan resultados a partir de lo que entro, si lo que recibe son datos “basura” pues nuestro modelo dará “basura”.

En primera instancia lo que se realizo fue una imputación de datos para poder controlar los datos nulos, para este punto se realizaron tres diferentes imputaciones por diferentes métodos los cuales fueron la media, mediana y KNN con 20 vecinos. El fin de realizar distintos métodos de imputación, es poder escoger el que menos se alejó de los datos originales. Para poder llevar a cabo lo anterior, se calculó un error cuadrado entre los tres diferentes métodos donde el que tuviera el valor más pequeño iba a ser nuestro método para usar, en este caso el que mejor funciono fue el método de imputación por la media.

Posterior a la imputación de datos y con ayuda de graficas que se realizaron en el EDA, se pudo observar que había varios atributos que poseían una distribución particular y que se tenían que analizar con detalle. Haciendo un estudio sobre estos atributos se llegó a la siguiente conclusión, duration y balance se debían normalizar, y pdays, campaign y previous se debían discretizar, esto se realizó pensando en los modelos que vamos a usar, ya que en algunos casos unos modelos trabajaban mejor con un rango de datos pequeños y que no hubiese negativos.

Por último, se consideró la opción de eliminar alguna columna que no propiciara ninguna información relevante, sin embargo, para este caso no se llegó a un ningún juicio concreto por lo que se decidió avanzar a la construcción de los modelos.

**4.2 Metodología**

Antes de empezar con el entrenamiento de los modelos, se realizó un Split sobre el dataset anterior en una proporción de 70/30 con el fin de que el 70% de los datos entrenen nuestros modelos y el 30% restantes prueben la eficacia de este. Esto se hace con el fin de poder evaluar de manera equitativa el rendimiento de los modelos que se trabajaran.

En primera medida se trabajó con Naive Bayes, y como se mencionó en un principio este método posee 4 variantes distintas, por lo que se tomó la decisión de entrenar un modelo de cada una y así poder evaluar entre estas su rendimiento. La forma en que se midió el rendimiento en este método fue con ayuda de la exactitud del modelo (accuracy\_score), donde a partir del mayor porcentaje era el modelo que confrontaría a los demás métodos que se trabajaran.

El segundo modelo que se trabajo fue Random Forest, donde se entrenó un modelo sencillo, es decir no se realizó ninguna optimización de los parámetros, y se midió su exactitud (accuracy\_score). Después se realizó una validación cruzada para ver si podíamos mejorar los resultados que se obtuvieron anteriormente, sin embargo, al finalizar esta parte se pudo observar que la exactitud del modelo no se vio afectada ni en el 1%, por esto mismo se tomó la decisión de continuar con el primero.

Con el último modelo supervisado, que fue Support Vector Machine, se realizó algo parecido como en Random Forest, se entrenó un modelo sin optimizar parámetros y posterior a esto se realizó una validación cruzada con el fin de poder obtener una mejor exactitud del modelo (accuracy\_score), en este caso si obtuvimos un mejor resultado con la validación cruzada, por lo tanto, se optó por tomar dicho modelo.

Pasando al modelo no supervisado (DBSCAN), se tomó el dataset original y se le elimino la columna de la clase, a partir de esto se creó y entreno el modelo con el dataset y se graficaron los resultados para ver con más claridad los diferentes clústeres generados.

Por último, se evaluaron a más detalle los modelos supervisados que se escogieron para ver que otra información más podemos obtener de estos, donde se miraron las siguientes métricas de, exactitud (accuracy\_score), precisión (precision\_score) y exhaustividad (recall\_score) del modelo. Luego de pensar que el dataset que manejamos es desbalanceado, ya que los casos que abundan más son los que no aceptan el crédito, y de haber hecho una investigación sobre dichas métricas, se llegó a la siguiente hipótesis, y es que el modelo no solamente tiene que ser capaz de clasificar correctamente si una persona quiere o no el crédito, sino que también debe tener en cierto sentido la habilidad de identificar a las personas que quieren el crédito. En primera instancia es difícil entender la diferencia entre clasificar e identificar, puesto que se entiende como sinónimos en algunos ámbitos, sin embargo, con ayuda de los resultados que analizaremos a continuación se podrá ver la diferencia que hay entre ambas cualidades.

**4.3 Resultados**

A continuación, mostraremos la matriz de confusión de cada modelo junto con métricas que se le pidieron a dicho modelo.

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

*Métricas y matriz de confusión para Naive Bayes*

Interfaz de usuario gráfica, Gráfico de rectángulos

Descripción generada automáticamente con confianza media

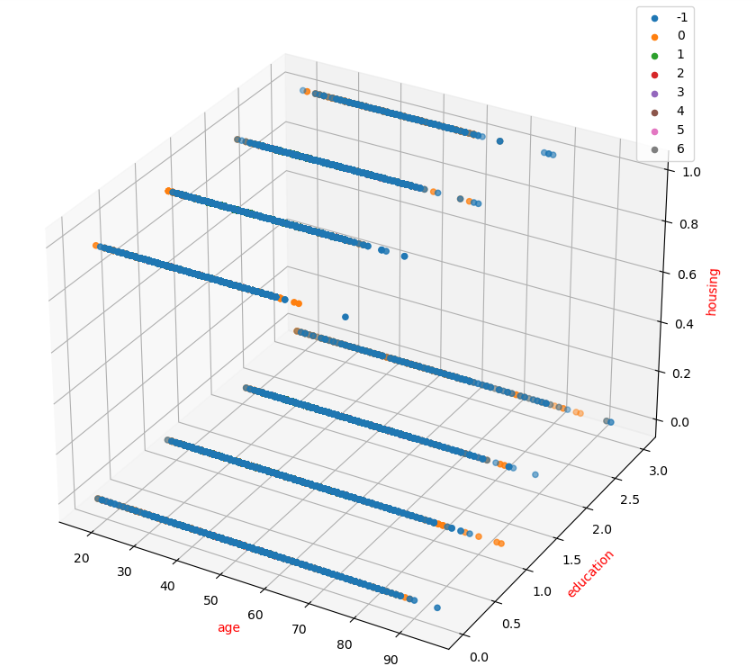
*Métricas y matriz de confusión para Random Forest*

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

*Métricas y matriz de confusión para Support Vector Machine*

Por último, mostraremos diferentes gráficos de lo que se obtuvo del método de clasificación no supervisada.



*Gráfico tridimensional que muestra la distribución de los clústers*

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

*Gráfico tridimensional, con age, month y balance en los ejes*

*Gráfico, Gráfico de dispersión

Descripción generada automáticamente*

*Gráfico tridimensional, que tiene como ejes day, month y balance*

En el método no supervisado no se puede apreciar muy al detalle los clústeres, más que todo porque nuestro dataset esta desbalanceado y esto hace que el modelo no genere tantos grupos.

**4.4 Discusión**

En este punto vamos a obviar al método de clasificación no supervisada, puesto que la hipótesis que planteamos anteriormente está dirigida a los modelos supervisados. Para entender un poco mejor de lo que se quiere llegar, se tiene que entender qué diferencia hay entre la exactitud (accuracy\_score), precisión (precision\_score) y exhaustividad (recall\_score) del modelo, y es que es muy común pensar que la exactitud ya determina que tan bueno es nuestro modelo, cuando en realidad esta métrica puede darnos malas aproximaciones si no se sabe el contexto de los datos. Antes de continuar con esta idea, daremos una explicación de cada métrica:

* Exactitud (accuracy\_score): Mide el porcentaje de casos que el modelo ha acertado a la hora de predecir.
* Precisión (precision\_score): Es un porcentaje que mide la calidad de la clasificación realizada por el modelo.
* Exhaustividad (recall\_score): Es un porcentaje que determina la cantidad que fue capaz el modelo de reconocer a las personas que si aceptaron.

Entonces más o menos tendiendo una idea de lo que representa cada métrica podemos observar que la exactitud (accuracy\_score), en nuestro caso no determina nada, esto sucede porque nuestro dataset tiene casi un 90% de personas que no aceptaron el crédito, lo que prácticamente hace que nuestro modelo no tenga problemas a la hora de clasificar a estas personas, y por consiguiente hará que la exactitud del modelo ronde entre los 80% y 90%. Ahora bien, el principal problema radica en saber cuándo el modelo fue capaz de identificar y a su vez de clasificar correctamente a la persona que acepto, es en este punto donde la precisión y exhaustividad del modelo entran en juego.

Cuando hablamos de la calidad de predicción, a lo que hacemos referencia es al porcentaje de personas que realmente aceptaron, sobre el total de personas que predijo el modelo que aceptaron, en otras palabras, si miramos el SVM tiene una precisión del 64%, es decir que el modelo cuando prediga que una persona está interesada tiene un 64% de estar en lo correcto, esto es bastante bueno, ya que después de todo es lo que buscamos en el modelo, que sea capaz de clasificar correctamente a una persona interesada.

Ahora bien, con lo anterior ya seriamos capaces de decir que SVM es el mejor modelo, sin embargo, este modelo tiene un problema grave y es el siguiente, de 10 personas que si aceptaron el crédito este modelo solo es capaz de identificar a 1 de ellas y esta estadística nos la da la exhaustividad. Para entender mejor el tema vamos a tomar a Naive Bayes de ejemplo, este modelo tiene una exhaustividad del 1%, lo que quiere decir que en el 99% de los casos donde las personas si aceptan el crédito, Bayes las tomara como personas que no lo aceptaron.

Este es un problema grave, ya que el banco busca que se acepten sus créditos, y si a esto le agregamos un modelo que limite aún más la adquisición por una mala predicción, lo que estaríamos haciendo es limitar las ganancias del banco por no identificar adecuadamente a la persona.

Ya entendiendo mejor lo que buscamos en el modelo vamos a hablar sobre Random Forest, este modelo tiene una exhaustividad bastante buena, ya que entre dos personas que acepten el crédito es capaz de reconocer a una, además de eso tiene un 62% de probabilidad de clasificar correctamente a la persona que acepto el crédito, sin embargo, este modelo tiene un problema y es que tiene una considerable cantidad de falsos positivos (personas que no aceptaron pero el modelo predijo que sí) y nos apoyaremos en la siguiente gráfica.

Gráfico

Descripción generada automáticamente

Como podemos ver SVM tiene una gran cantidad de falsos negativos (personas que aceptaron, pero que el modelo no las identifico) y Random Forest disminuye considerablemente este pico, pero como se observa en falsos positivos (personas que no aceptaron, pero el modelo predijo que sí) el Random Forest duplica aproximadamente las personas a comparación de SVM, y en verdaderos positivos (personas que aceptaron y el modelo predijo que sí) se puede observar que Random Forest casi triplica la cantidad de personas, ya que fue capaz de identificar más.

Llegados a este punto, tenemos que analizar para que se pensó el modelo, y es que, al iniciar una campaña con un grupo de personas, se quiere predecir cuales van a ser las que seguramente acepten el crédito, claramente pensando en que las campañas representan costos a los bancos y si aseguramos que estas sean exitosas, entonces se estaría maximizando las ganancias obtenidas. Ya para concluir con el tema lo que se le recomendaría al banco seria Random Forest, puesto que lo que más importa es que este modelo es capaz de reconocer mejor a las personas que si están interesadas, a diferencia de SVM que descarta una gran cantidad de personas que probablemente si hubieran aceptado el crédito.

1. **Conclusiones**
   1. Algo importante que podemos ver de la discusión es que no podemos simplemente ingresar datos y automáticamente ya tenemos el mejor modelo, si no que tiene que haber un análisis de fondo donde la persona que está encargada de interpretar dichos datos sea capaz de dar la mejor respuesta al problema considerando diferentes aspectos que sean relevantes en la situación del caso.
   2. Como se pudo observar el ejemplo que se elaboró a lo largo del documento, era un caso de datos reales ya que fueron recolectados de un banco portugués, a lo que se quiere llegar con todo esto es que hacer entender que la minería de datos no es ajena al mundo que nos rodea, y de fondo muchas implementaciones como la que acabamos de hacer son usadas para disminuir gastos innecesarios o aumentar la rentabilidad obtenida en una empresa, entre muchas otras aplicaciones.
   3. Por último, hay que mencionar que ningún modelo de predicción es mejor que otro, en nuestro ejemplo el que mejor funciono fue Random Forest, pero en otros casos pudo haber sido otro. Cada modelo tiene sus cualidades que deben ser estudiadas para poder determinar en qué momento es mejor sobre el caso que se está manejando.

# **Bibliografía**

Heras, J. M. (09 de 10 de 2020). *IArtificial.net*. Obtenido de IArtificial.net: https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/

Jauregui, A. F. (s.f.). *DBSCAN in Python: learn how it works*. Obtenido de DBSCAN in Python: learn how it works: https://anderfernandez.com/en/blog/dbscan-python-tutorial/

Rodrigo, J. A. (Diciembre de 2020). *Máquinas de Vector Soporte (SVM) con Python*. Obtenido de cienciadedatos.net: https://www.cienciadedatos.net/documentos/py24-svm-python.html

Rodrigo, J. A. (Octubre de 2020). *Random Forest con Python*. Obtenido de cienciadedatos.net: https://www.cienciadedatos.net/documentos/py08\_random\_forest\_python.html

*uci.edu*. (2014). Obtenido de Bank Marketing Data Set: https://archive.ics.uci.edu/ml/datasets/Bank+Marketing#

Diagrama

Descripción generada automáticamente