**Rubén Hernández Guevara**

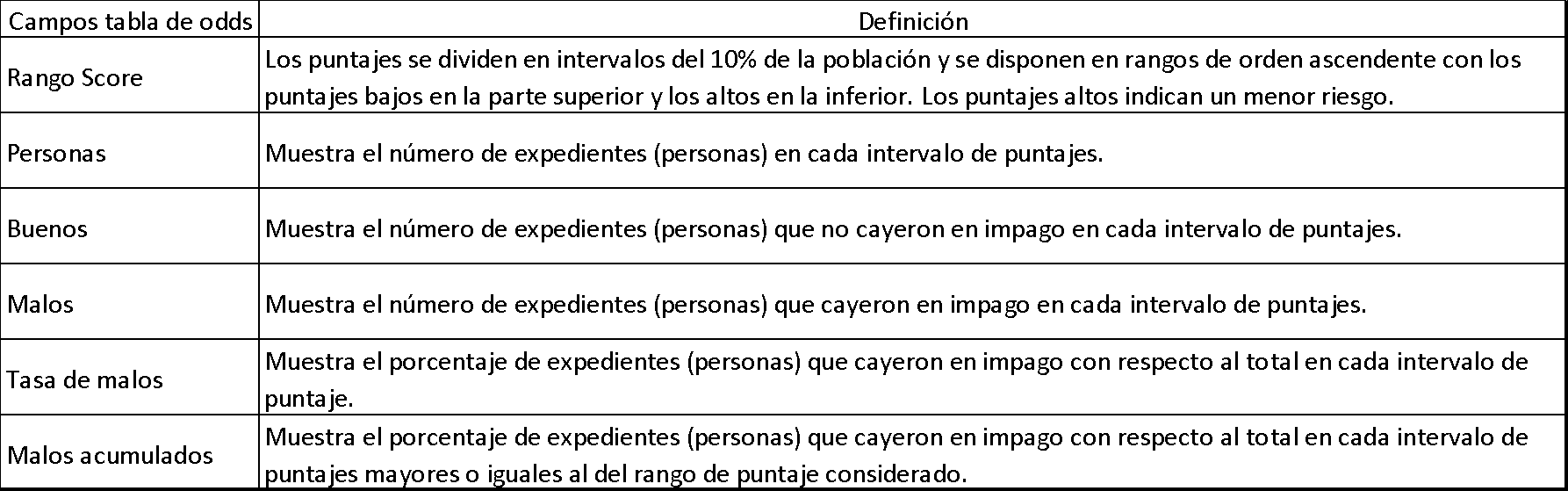
**Repositorio Proyecto:** [**https://github.com/Ruhguevara/challenge\_cdc**](https://github.com/Ruhguevara/challenge_cdc)

**Carpeta Drive:** [**https://drive.google.com/drive/folders/1HN0uV-YV8Yyo4FJ04ex3OoqIbAVmua0t?usp=sharing**](https://drive.google.com/drive/folders/1HN0uV-YV8Yyo4FJ04ex3OoqIbAVmua0t?usp=sharing)

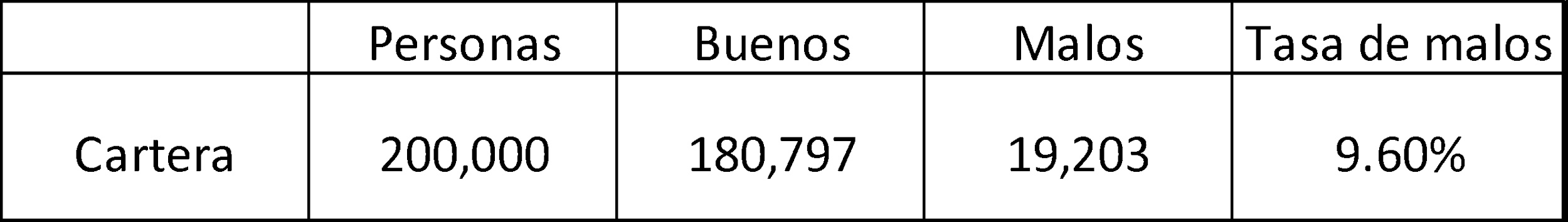
Contesta este examen de manera individual. Puedes buscar información en sitios web si así lo requieres. Nos gustaría conocer cómo abordas los problemas. ¡Éxito!

**Conocimientos de Riesgo de Crédito.**

1. **¿Cuáles son las etapas en el ciclo del crédito?**
   1. **Solicitud de Crédito**: La persona o empresa solicita un crédito y brinda información.
   2. **Evaluación de la Solicitud**: La institución evalúa la información brindada por el cliente y la viabilidad del crédito.
   3. **Decisión**: Con la información procesada y con potencial resultado favorable o no, la institución comunica al cliente la decisión.
   4. **Formalización**: Se establecen formalmente los términos y condiciones del crédito y se formaliza mediante un contrato.
   5. **Seguimiento/gestión**: Se brinda seguimiento constante al cliente informando saldos restante, pagos, notificaciones, recordatorios, etc.
   6. **Amortización/Liquidación:** Una vez que se han realizado todos los pagos requeridos, el crédito se considera pagado en su totalidad.
2. **Menciona algún indicador de riesgo de crédito.**
   1. **Score Crediticio**: Puntaje cuyo objetivo es reflejar la capacidad del cliente para cumplir con sus obligaciones financieras.
   2. **Capacidad de pago:** Evaluación de la capacidad del solicitante para hacer frente a los pagos futuros según sus ingresos y obligaciones.
3. **Supongamos que un banco quiere desarrollar un modelo de score que le indique qué personas que solicitan un crédito van a caer en impago y cuáles no. Dicho banco desarrolló dos modelos, el ordenamiento de cada modelo se muestra en las siguientes tablas:**

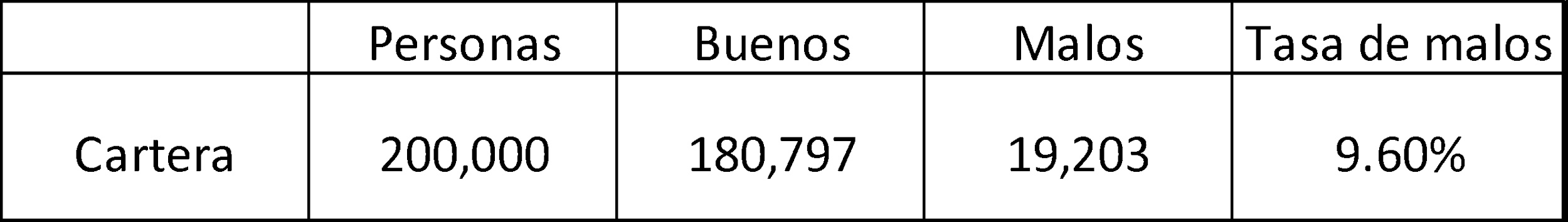
Definiciones: 

Modelo 1





Modelo 2:





Justifica cuál de los dos modelos funciona correctamente.

**Insights:**

**Hay una pequeña inconsistencia en los datos de los modelos:**

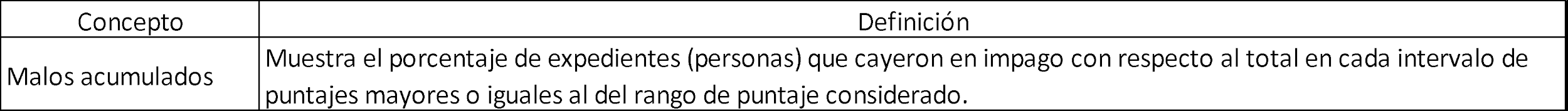
* **modelo 1: ‘buenos’ = 180,800. ‘malos’ = 19,203. Total = 200,003.**
* **modelo 2: ‘buenos’ = 180,185. ‘malos’ = 19,815. Total = 200,000.**
  + **Esto hace que la tasa de ‘malos’ en realidad sea de 9.91%**

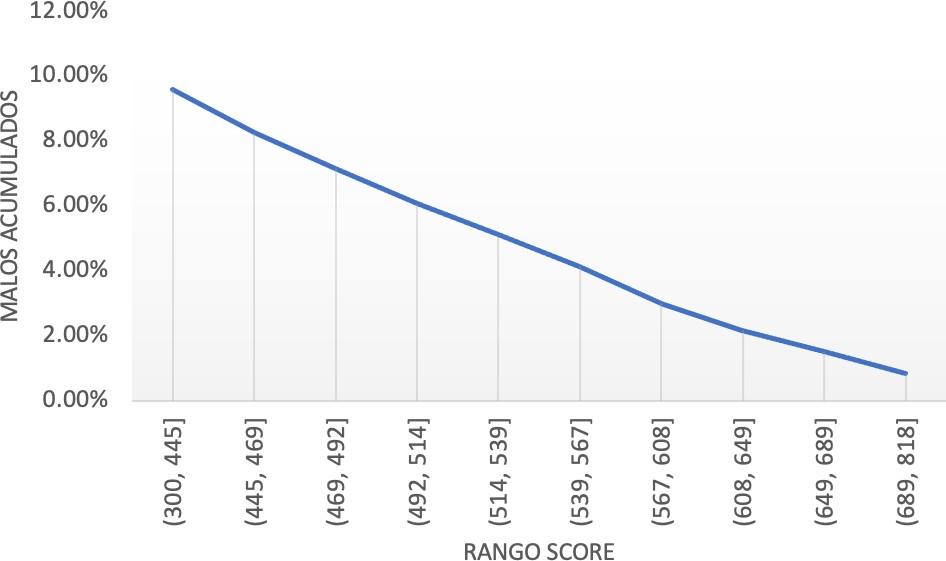
El modelo 1 es mejor, en los últimos 3 deciles (30% de la población) existe un porcentaje de malos acumulados de .65%, mientras que para el modelo 2 es de .75%, y si bien la diferencia es poca, el modelo 2 tiene problema con la fuga de ‘malos’ a partir del decil 3, estos empiezan a incrementarse en lugar de seguir una tendencia a la baja, lo cual indica que algo no está funcionando tan bien como en el modelo 1, ya que este muestra una relación coherente entre las variables y captura adecuadamente el riesgo crediticio a medida que aumenta el rango de score.

En el modelo 2, la relación entre las variables no es coherente. A medida que aumenta el rango de score, los "Malos Acumulados" no disminuyen de manera consistente, e incluso aumentan en algunos rangos. Además, la "Tasa Malos" también fluctúa sin un patrón claro a lo largo de los rangos de score.

[scorecards.xlsx](https://docs.google.com/spreadsheets/d/1Pra7XQkJqu9aSyWB6oCGKwoFEBKlFtfw/edit?usp=sharing&ouid=110416587076795433783&rtpof=true&sd=true)

4. Considerando la gráfica que se muestra a continuación, ¿qué estrategia de aceptación puede implementar el banco si su apetito de riesgo es que no más del 5% de su cartera caiga en impago?





**Respuesta:**

* Para implementar una estrategia de aceptación que mantenga el riesgo de impago en no más del 5%, el banco debería establecer un umbral de riesgo máximo para la aceptación de créditos. El umbral se puede establecer en función de la tasa de impago acumulada del modelo de score y el apetito de riesgo del banco. En este caso, el banco no quiere que su cartera caiga en impago más del 5%, por lo que el umbral de riesgo máximo podría ser del 5%. El punto de corte puede ser en 514.
* Se puede crear también una función de optimización, en donde se pueda optimizar el punto de corte, que minimizaría la diferencia entre el % de malos y la meta del banco (0.05), así se obtendría un score óptimo.

**Conocimientos de Python y SQL**

1. **En un dataframe (df), ¿cuál es la sintaxis para obtener el total de na’s para cada variable?**
   * Los dataframes tienen los métodos ***.isnull()*** *e* ***isna()***, que regresan True o False en caso de que un valor sea ***‘na’***; *computacionalmente, los True y False son unos y ceros*, es por eso que se le puede aplicar en cadena el método ***.sum()***, que hace la sumatoria de estos unos y ceros, para así obtener el total de na’s para cada variable.

| df.isnull().sum() |
| --- |

1. **En ese mismo df, ¿cuál es la sintaxis para imputar los valores perdidos na con el promedio? Supón que los valores perdidos se encuentran en la variable ‘edad’ y ‘nivel\_socieco’.**

| df[‘edad’] = df[‘edad’].fillna(df[‘edad’].mean())  y  df['nivel\_socieco'] = df['nivel\_socieco'].fillna(df['nivel\_socieco'].mean()) |
| --- |

1. **Crea un histograma a través de la librería seaborn para la variable edad.**

Adicionalmente me gusta utilizar el parámetro **‘kde’,** el cual muestra el ‘kernel density estimate’, que es una suavización de la distribución de los datos.

**‘hue’** para visualizar variables categóricas en el histograma.

**‘bins’** para tener control de la cantidad de ‘barras’ del histograma.

**‘stat’** para mostrar cierto tipo de estadístico en el eje y.

entre otros.

| sns.histplot(data=df, x='edad', kde=True, hue = ‘sexo’, bins = np.max(df[‘edad’]), stat =..., )  plt.title('Distribución de Edades')  plt.xlabel('Edad')  plt.ylabel('Frecuencia')  plt.show() |
| --- |

1. **Estás a punto de preparar un modelo y para ello quieres separar en entrenamiento y validación. ¿Con qué sintaxis separarías tu base en un 70% para entrenamiento y el restante para validación? ¿qué librería usarías? Esto con una semilla aleatoria.**

42: ‘The Answer to the Ultimate Question of Life, The Universe, and Everything’.

| from sklearn.model\_selection import train\_test\_split  X\_train, X\_val, y\_train, y\_val = train\_test\_split(X, y, test\_size=0.3, random\_state=42) |
| --- |

1. **Si quisieras implementar un Random Forest 10 árboles y una máxima profundidad de 2. ¿Cuál sería la sintaxis? Puedes apoyarte de lo sig:**

# Librería a importar

from sklearn import ….

# Inicializar el modelo

rf = …..(n\_estimators = …..)

# Entrenar el modelo en tu base de entrenamiento: x\_train, y\_train

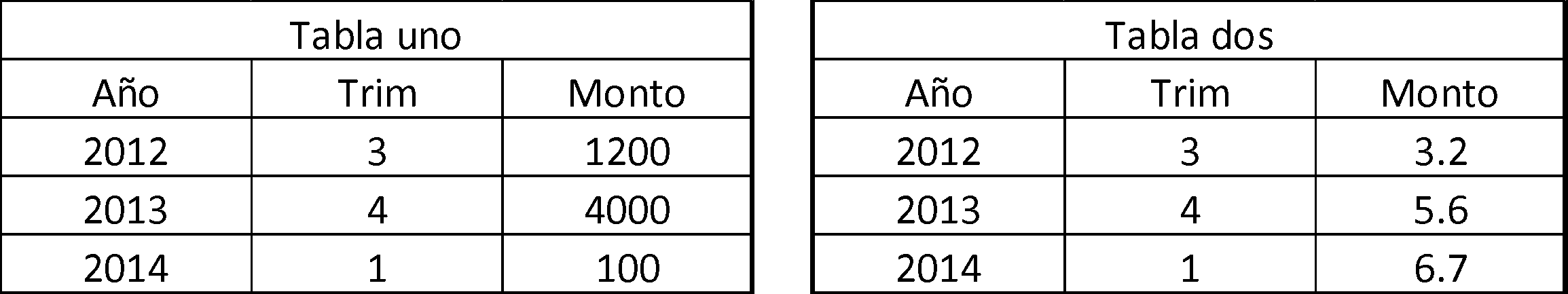
rf.fit(…. , ….)

| from sklearn.ensemble import RandomForestClassifier  rf = RandomForestClassifier(n\_estimators=10, max\_depth=2)  rf.fit(X\_train, y\_train) |
| --- |

1. **Si quisieras verificar algunas métricas sobre tu modelo de Random Forest, ¿qué librería y sintaxis usarías para graficar la curva ROC? Supón que obtienes un valor de 0.53 de AUC, ¿este valor te diría que tu modelo es preciso?**

| from sklearn.metrics import roc\_curve, roc\_auc\_score  import matplotlib.pyplot as plt  predictions = random\_forest.predict\_proba(X\_test)[:, 1]  labels = y\_test  fpr, tpr, thresholds = roc\_curve(labels, predictions)  auc = roc\_auc\_score(labels, predictions)  # Graficar la curva ROC  plt.plot(fpr, tpr, label='ROC curve (AUC = {})'.format(auc))  plt.xlabel('False Positive Rate')  plt.ylabel('True Positive Rate')  plt.show()  !!Un AUC de .53 es relativamente bajo, y el modelo no es muy preciso!! |
| --- |

1. **Supón que tienes la sig información en las tablas uno y dos:**



Escribe código sql para hacer un inner join de las tablas y seleccionar la suma de monto \* precio por año.

| SELECT t1.Año, SUM(t1.monto \* t2.precio) AS total, FROM tabla1 AS t1 INNER JOIN tabla2 AS t2 USING (año) GROUP BY t1.Año; |
| --- |

**Conocimientos de Ciencia de Datos**

1. **¿Qué significan las siguientes curvas de aprendizaje?**

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

1. El modelo está sobre ajustado (overfit).
2. El modelo está sub ajustado (underfit).
3. El modelo está bien entrenado.
4. Ninguna de las anteriores.

Las gráficas de curvas de aprendizaje muestran el desempeño en el tiempo, el eje-y es alguna métrica de aprendizaje (classification accuracy, por ejemplo), y el eje-x es la ‘experiencia’ (tiempo).

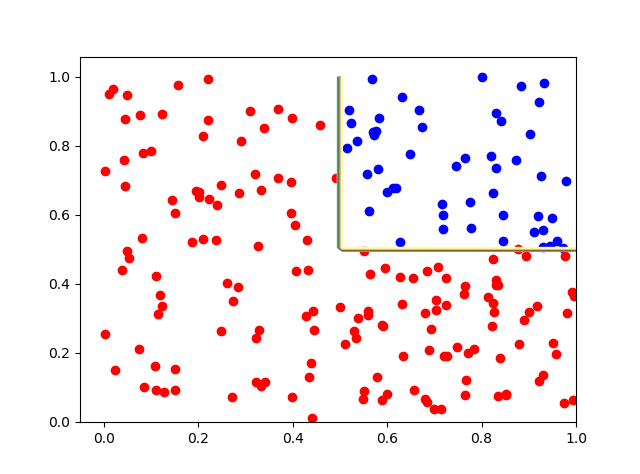
En la gráfica, a medida que el número de iteraciones aumenta, el error de training está mejorando, mientras que el error de validación no, de hecho, está incrementando levemente. Esto es una clara señal de overfitting y de que el entrenamiento puede ser detenido para no consumir demasiados recursos computacionales (y evitar posible desgaste de recursos $).

1. Supón que tu modelo está sobre ajustado. ¿Qué harías para mejorar eso?

Hay varias maneras de tratar esto:

* Regularización (reducir qué tan sensible es el modelo a los datos de entrenamiento, incrementa el bias, pero a cambio, reduce la varianza)
  + L1 (lasso)
  + L2 (ridge)
* Selección de features
* Cross validation
* Re-considerar un modelo no tan complejo que balancee Bias-Variance.

1. ¿Qué tipo de algoritmo realiza una separación como la que se muestra en la figura?



1. Máquina de Soporte Vectorial con Kernel Lineal
2. Regresión Logística
3. Árbol de Decisión
4. Random Forest
5. **¿Qué es el aprendizaje no supervisado? Menciona un ejemplo.**

El aprendizaje no supervisado es una rama de Machine Learning en la cual no se proporcionan etiquetas o respuestas correctas para entrenar un modelo, el objetivo principal es descubrir patrones, estructuras o relaciones en los datos, no hay un “target”. Agrupa e interpreta datos basado unicamente en los datos input.

Ejemplos:

* Clustering
  + K-Means
  + GMM
* Reducción de Dimensiones
  + PCA
  + t-SNE
  + UMAP

1. **Si se te pidiera elaborar un modelo de machine learning de clasificación para un problema de fraude que tiene una bad rate del 2.0%, ¿con qué tipo de problema te podrías estar enfrentando? ¿cómo lo solucionarías? ¿qué tipo de modelo de ML usarías?**

Uno de los principales problemas es el desbalance de clases, si se trabajara sin aplicar alguna técnica como Oversampling, Subsampling o Asignación de Pesos a las clases, el modelo sería algo ineficiente al momento de reconocer patrones de la clase minoritaria.

* En el mundo real, buscaría una muestra de datos más grande, para incrementar de manera ‘orgánica’ la clase minoritaria, luego, utilizaría alguna técnica como SMOTE, Upsampling, o Asignación de Pesos.
* En cuanto al modelo, me gusta siempre utilizar primero modelos más sencillos pues ayuda a dar una vista general de la complejidad del problema, si el modelo sencillo da relativos buenos resultados, es señal de que quizá no es tan complejo e incluso no sea necesaria la implementación de modelos más avanzados como Redes Neuronales, que, computacionalmente son más pesados y a veces difíciles de interpretar. El punto es descubrir la complejidad del problema sin utilizar tantos recursos e incrementar la complejidad sin algo a cambio.
* En este caso, utilizaría primero la Regresión Logística para ver qué tanto es capaz de generalizar. Dependiendo de esto, utilizaría luego otro modelo más avanzado, como Gradient Boosting o Árboles de Decisión.

1. **Cuál consideras tu que una métrica de performance es buena para un problema desbalanceado, es decir, 60% de AUCE es una métrica buana o mala y porqué.**

**De mi notebook:**

* Tener un Recall alto significa la detección de más casos verdaderos de fraude, optimizar aumenta TP y reduce FN, pero incrementa FP, esto significa, más acusaciones incorrectas de fraude, lo que puede perjudicar la satisfacción del cliente y la reputación de la institución, pero al mismo tiempo, se evitan pérdidas debido a detección de fraudes.
* Tener una Precisión alta, significa más seguridad de que el caso es verdaderamente fraudulento. Esto implica menos acusaciones incorrectas, pero también puede haber más pérdidas debido a casos que no fueron detectados.
* La decisión final es a consideración de la institución, yo he decidido ponderar igualmente ambas métricas, por lo que continuaré con los resultados de la optimización de f1 para el análisis de deciles.

En el caso de la AUC, en pocas palabras significa qué tan bien el modelo separa las clases, mientras más alto su valor (0-1), mejor. Un AUC de .60 es relativamente bajo, y en este contexto, no lo utilizaría en producción, buscaría mejorar el modelo o cambiarlo, pues es crucial lograr buenos resultados.

**Problema en Python:**

La empresa necesita realizar un modelo de fraude. Tu papel como Científico de datos es dar una solución de modelo y mitigar los riesgos de fraude basada en datos. Los datos para realizar el modelo son datos.csv.

Sugerencia de pasos a considerar en tu modelo de fraude:

* Análisis exploratorio de los datos.
* Preparación de los datos.
* Selección del modelo (¿con qué tipo de problema te estás enfrentando?).
* Verificación del mismo y sus métricas.
* Conclusiones.

Entregables:

* Jupyter notebook con el código
* Hallazgos de tu investigación (Puede ser en el mismo notebook)

Recuerda que en un contexto de trabajo en equipo, las personas que leerán tu código puede que no estuvieron involucradas en su desarrollo pero igual tendrán que entendenderlo y/o mantenerlo. Además deberás comentar cada sección y compartir tus hallazgos.

Datos:

**id:** Identificador único de la transacción

**timestamp:** momento en el que ocurre la transacción

**amount:** monto de la transacción

**variables\_01 a variable\_32:** features de la trasacción

**fraud:** Flag de flaude 1 es transacción fraudulente y 0 no lo es