Reporte: Predicción de Churn de Clientes

-Estuardo García 20220112

1. Introducción

En un mercado cada vez más competitivo, retener clientes es fundamental para asegurar la rentabilidad y el crecimiento de la empresa. Este proyecto se centra en **predecir el churn** (abandono de clientes) mediante técnicas de Machine Learning, permitiendo identificar a aquellos clientes con mayor riesgo de dejar de utilizar nuestros servicios. Con esta información, el área de marketing y atención al cliente podrá implementar estrategias proactivas para mejorar la retención y reducir los costos asociados a la pérdida de clientes.

2. Objetivos del Proyecto

- **Reducir la tasa de churn:** Identificar a los clientes en riesgo para aplicar acciones de retención específicas.
- **Optimizar recursos:** Enfocar las campañas de marketing y retención en los clientes que realmente lo requieren, evitando el gasto innecesario.
- **Mejorar la satisfacción del cliente:** Al anticipar posibles cancelaciones, se pueden ajustar los servicios y la comunicación para aumentar la fidelidad.

3. Descripción de las Variables

1. **Age:**
   La edad del cliente en años.
2. **Gender:**
   El género del cliente (por ejemplo, Male, Female).
3. **Usage Frequency:**
   La frecuencia mensual con la que el cliente utiliza el servicio.
4. **Support Calls:**
   El número de llamadas de soporte realizadas por el cliente.
5. **Payment Delay:**
   El número de días que el cliente se retrasó en el pago.
6. **Subscription Type:**
   El tipo de plan de suscripción al que el cliente está suscrito (por ejemplo, Basic, Standard, Premium).
7. **Contract Length:**
   La duración del contrato de suscripción (por ejemplo, Monthly, Quarterly, Annual).

8. **Total Spend:**
   El monto total gastado por el cliente durante su relación con la empresa.
9. **Last Interaction:**
   El número de días transcurridos desde la última interacción del cliente con el servicio.
10. **Churn:**
    Variable binaria que indica si el cliente ha abandonado el servicio (1 para churn, 0 para retención).

4. Análisis exploratorio de datos

1. Se agruparon las variables en categorías y se calculó la tasa de churn promedio para cada grupo. Luego, se graficaron los resultados en diagramas de barras, lo que permitió observar cómo la probabilidad de abandono variaba según los diferentes rangos o categorías de las variables analizadas.

IMAGENES PROYECTO **(Link con las gráficas)**

5. Tratamiento de los datos

1. **Age:** Se aplicó MinMaxScaling para normalizar la variable en un rango de 0 a 1, eliminando diferencias de escala que pudieran afectar el análisis.

2. **Gender:** Se utilizó LabelEncoder para transformar esta variable categórica en numérica, asignando 0 a "Female" y 1 a "Male".

3. **Usage Frequency:** Se normalizó mediante MinMaxScaling para ajustar sus valores a una escala uniforme y minimizar el impacto de posibles valores atípicos.

4. **Support Calls:** Se aplicó MinMaxScaling para estandarizar el número de llamadas de soporte, facilitando su comparación con otras variables.

5. **Payment Delay:** Se escaló usando MinMaxScaling para normalizar los días de retraso en el pago, permitiendo un análisis más homogéneo.

6. **Subscription Type:** Se transformó a valores numéricos mediante LabelEncoder, asignando 0 a "Basic", 1 a "Standard" y 2 a "Premium", lo que permite tratar la información categórica de forma numérica.

7. **Contract Length:** Se convirtió la variable a meses y posteriormente se aplicó MinMaxScaling para normalizar la duración del contrato, garantizando que la variable esté en una escala comparable.

8. **Total Spend:** Se normalizó con MinMaxScaling para reducir el efecto de valores extremos y asegurar una escala uniforme en el gasto acumulado.

9. **Monthly_Spend:** Se generó mediante ingeniería inversa al dividir la variable Total Spend entre Tenure, obteniendo así el gasto mensual promedio de cada cliente, y luego se normalizó con MinMaxScaling.

10. **Last Interaction:** Se aplicó MinMaxScaling para estandarizar la variable que indica los días desde la última interacción, facilitando su análisis en conjunto.

11. **Churn:** Se mantuvo como variable binaria sin transformación adicional, representando 0 para retención y 1 para abandono.

- **Conclusión:** se aplicó un tratamiento especializado a cada variable para asegurar la homogeneidad y la compatibilidad en el análisis. Se normalizaron las variables numéricas mediante MinMaxScaling para eliminar diferencias de escala y reducir el impacto de valores atípicos, mientras que las variables categóricas se transformaron en formatos numéricos a través de técnicas de codificación como LabelEncoder. Además, se generó la variable Monthly_Spend mediante ingeniería inversa para reflejar el gasto mensual promedio, y se ajustaron las variables derivadas, como Contract Length, convirtiéndolas a una unidad común (meses). Este enfoque integral permitió preparar un dataset limpio y estructurado, optimizado para la construcción y evaluación de modelos predictivos.

6. Modelo: SGDClassifier

Configuración y Entrenamiento

- Se instanció un SGDClassifier con loss='log_loss', alpha=0.001, random_state=42 y max_iter=5000. El modelo se ajustó a los datos de entrenamiento para aprender los parámetros óptimos que minimizan la función de pérdida.

Evaluación Inicial (Umbral por Defecto)

- Se generaron predicciones en el conjunto de prueba y luego se calculó la exactitud ‘’accuracy score’’ y se obtuvo el ‘’Classification Report’’, obteniendo métricas como precisión, recall y F1-score. Esto sirvió como línea base para entender el desempeño con el umbral estándar de 0.5.

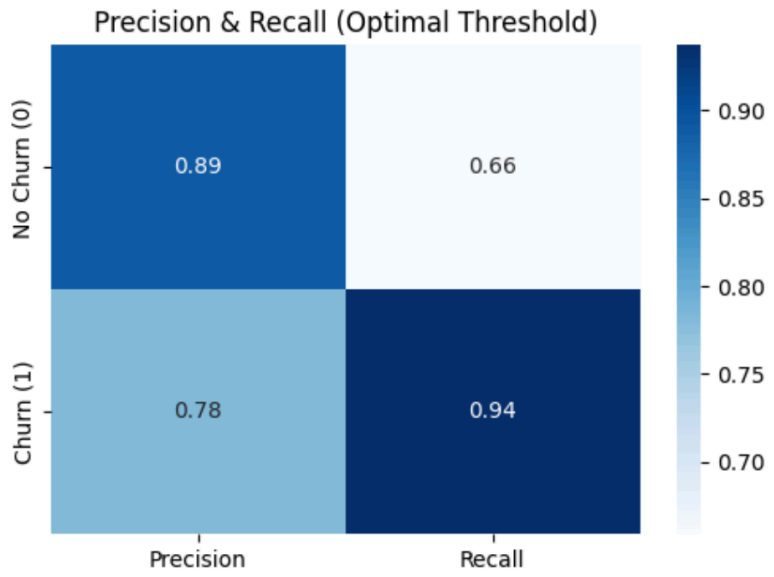
Optimización de Umbral según Matriz de Costos

- Para considerar el impacto económico de diferentes tipos de error, se definió un costo de ‘’1’’ para los falsos positivos y ‘’3’’ para los falsos negativos.
- Se recorrió un rango de umbrales desde 0 a 1 (en pasos de 0.01), calculando en cada punto el costo total de FP y FN sobre el conjunto de prueba.
- El umbral óptimo se identificó como aquel que minimiza el costo total.

Evaluación con Umbral Óptimo

- Se recalcularon las predicciones usando el umbral que minimiza el costo. Luego se midió la exactitud y se imprimió un Classification Report actualizado, reflejando las nuevas métricas de desempeño con este umbral personalizado. Esta estrategia permitió ajustar el modelo a la realidad de negocio, reduciendo el impacto de errores más costosos ‘’falsos negativos’’ a cambio de tolerar potencialmente un mayor número de falsos positivos.

```
SGDClassifier Accuracy (default threshold): 0.8570553608492973
SGDClassifier Classification Report (default threshold):
              precision    recall  f1-score   support

         0.0       0.82      0.85      0.84     38167
         1.0       0.88      0.86      0.87     50000

    accuracy                           0.86     88167
   macro avg       0.85      0.86      0.85     88167
weighted avg       0.86      0.86      0.86     88167

Optimal threshold based on cost matrix: 0.3
SGDClassifier Accuracy (optimal threshold): 0.8166774416731883
SGDClassifier Classification Report (optimal threshold):
              precision    recall  f1-score   support

         0.0       0.89      0.66      0.76     38167
         1.0       0.78      0.94      0.85     50000

    accuracy                           0.82     88167
   macro avg       0.84      0.80      0.80     88167
weighted avg       0.83      0.82      0.81     88167
```



Análisis de Métricas del SGDClassifier

- **Exactitud:** Se ubica en 0.82, ligeramente menor que con el umbral por defecto (0.86), reflejando el compromiso entre exactitud global y detección de clientes en riesgo.
- **Clase 0 (No Churn):**
  - Precisión: 0.89
  - Recall: 0.66
  - F1-Score: 0.76
    Aun cuando la precisión se mantiene alta, la menor capacidad de detección (recall) para esta clase indica que el modelo tiende a clasificar más clientes como Churn.
- **Clase 1 (Churn):**
  - Precisión: 0.78
  - Recall: 0.94
  - F1-Score: 0.85
    El recall elevado (0.94) implica que el modelo identifica a la mayoría de los clientes que realmente abandonan, alineándose con el objetivo de reducir el costo de no detectar casos de churn.

7. Modelo: KNeighborsClassifier

Configuración y Entrenamiento

- Se definió un `KNeighborsClassifier`, especificando un número de vecinos ‘’k=5’’ que se determinó a través de experimentación o validación cruzada. Luego se ajustó el modelo con los datos de entrenamiento, calculando las distancias entre cada muestra y sus vecinos más cercanos para determinar la clase predicha.

Evaluación Inicial (Umbral por Defecto)

- Se obtuvieron predicciones sobre el conjunto de prueba, después se calculó la el ‘’accuracy’’ y se generó el Classification Report, evaluando precisión, recall y F1-score para la clase churn y la clase no churn. Esto sirvió como referencia inicial del rendimiento con el umbral estándar (0.5)

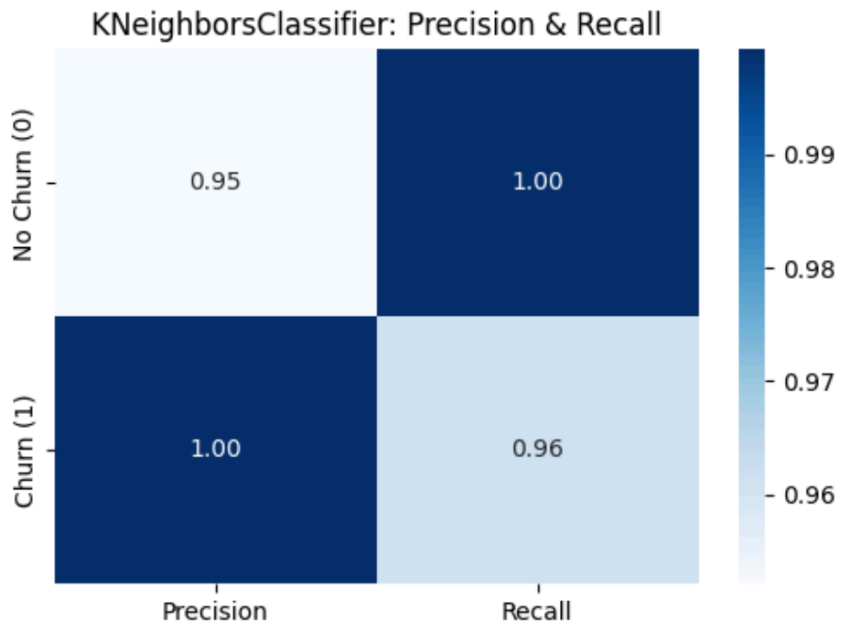
Optimización de Umbral según Matriz de Costos

- Dado que `este modelo` permite obtener probabilidades se exploró un rango de umbrales (por ejemplo, de 0 a 1 en incrementos de 0.01).
- Para cada umbral, se clasificó a los clientes en churn o no churn, calculando el costo total en función de la matriz de costos ‘’1 para falsos positivos y 3 para falsos negativos’’.
- Se identificó el umbral óptimo como aquel que minimiza la suma ponderada de errores, reduciendo la penalización de los falsos negativos es decir los clientes en riesgo que no se detectan).

Resultados

- Con el umbral por defecto, el modelo alcanzó un desempeño equilibrado entre exactitud y recall para la clase churn. Al ajustar el umbral según la matriz de costos, la exactitud general pudo disminuir ligeramente, pero se incrementó la detección de clientes con mayor probabilidad de abandono, reduciendo así el costo asociado a no intervenir a tiempo. Este enfoque personalizado de umbral permitió al KNN enfocarse en la retención de los clientes más propensos a churn, alineando el modelo con los objetivos del modelo

```
KNeighborsClassifier Accuracy: 0.9778261707895244
KNeighborsClassifier Classification Report:
              precision    recall  f1-score   support

         0.0       0.95      1.00      0.98     38167
         1.0       1.00      0.96      0.98     50000

    accuracy                           0.98     88167
   macro avg       0.98      0.98      0.98     88167
weighted avg       0.98      0.98      0.98     88167
```



Análisis de Métricas del KNeighborsClassifier:

- **Clase 0 (No Churn):**
  - Precisión: 0.95
  - Recall: 1.00
  - F1-Score: 0.98
    La recall perfecta indica que no se dejan escapar clientes que realmente se quedan en la empresa, mientras que la alta precisión evita falsos positivos en esta clase.
- **Clase 1 (Churn):**
  - Precisión: 1.00
  - Recall: 0.96
  - F1-Score: 0.98
    El modelo identifica con total precisión los casos de churn que predice como tal, y su recall de 0.96 sugiere muy pocos falsos negativos.

7. Conclusion

En conclusión, se desarrolló un análisis integral que abarcó desde la definición del caso de negocio y la descripción detallada de las variables, hasta un exhaustivo tratamiento de datos que incluyó técnicas de normalización, codificación e ingeniería inversa. Se implementaron dos modelos predictivos ‘’SGDClassifier y KNN’’ ajustando los umbrales de decisión mediante una

matriz de costos para minimizar el impacto económico de los errores más críticos. Los resultados obtenidos demuestran que, a pesar de una ligera disminución en la exactitud general al optimizar el umbral, se logró una mejora significativa en la detección de clientes en riesgo de churn, lo cual se alinea con los objetivos estratégicos de retención y optimización de recursos de la empresa.