SPEED DATING DATASET

Por:

José Franco Arroyave

Isaac Esteban Uribe

Profesor: Raul Ramos Pollan

Universidad de Antioquia

16/11/2023

Introducción

Este informe tiene como objetivo presentar un resumen completo de nuestro proyecto de Machine Learning, el cual se desarrolló en el contexto de las citas y la mejora de una plataforma de recomendaciones para usuarios. Durante el proceso de desarrollo, hemos trabajado en la exploración, el preprocesamiento de datos, la implementación de modelos supervisados y no supervisados, y el análisis exhaustivo de métricas de desempeño. A lo largo de este informe, se detallan los avances clave y los desafíos que hemos enfrentado.

En la fase inicial de nuestro proyecto, nos sumergimos en la exploración del conjunto de datos de citas rápidas proporcionado por la Columbia Business School a través del portal Kaggle. Nuestro objetivo era comprender a fondo las 195 variables disponibles, sus significados y su relevancia en el contexto de las citas rápidas. A medida que avanzamos, realizamos una selección de características, eliminando variables redundantes y preparando los datos para el modelado.

Nuestras iteraciones de desarrollo se centraron en la implementación de modelos supervisados y no supervisados. Evaluamos varios modelos de clasificación y técnicas de agrupamiento para refinar nuestra solución. Realizamos pruebas de métricas de desempeño, como precisión y recall, con resultados prometedores en cuanto a la precisión de los modelos. Sin embargo, el recall demostró ser un área de mejora.

El despliegue de este proyecto implica consideraciones críticas, como la escalabilidad de la infraestructura, la privacidad de los datos y la comunicación efectiva de resultados.

Resumen

Este informe ejecutivo presenta los hallazgos y resultados de un proyecto de Machine Learning que se centró en predecir si una pareja que se conoce en una cita rápida de cuatro minutos volverá a verse. El proyecto utilizó un conjunto de datos de citas rápidas de la Columbia Business School que contiene información sobre los participantes en eventos de citas rápidas entre 2002 y 2004. El objetivo del proyecto es proporcionar recomendaciones precisas sobre la compatibilidad de las parejas, lo que mejora significativamente la experiencia de los usuarios en la aplicación de citas.

Carga de los Datos

El proceso de carga de los datos del conjunto de citas rápidas de la Columbia Business School marca el inicio del proyecto. Esta etapa reviste gran importancia, ya que proporciona una visión inicial de la calidad y la estructura de los datos. Los datos se extrajeron del portal web Kaggle y se cargaron en el entorno de trabajo de análisis de datos. Durante este proceso, se revisó la integridad de los datos, asegurando que no hubiera problemas de formato o lectura.

El dataset en cuestión contenía un total de 195 variables, lo que implica una gran cantidad de información para comprender. Cada variable se sometió a un análisis inicial para determinar su significado, su tipo de dato y su potencial contribución al objetivo del proyecto. Esta exploración permitió establecer una base sólida para la posterior toma de decisiones en términos de preprocesamiento y modelado.

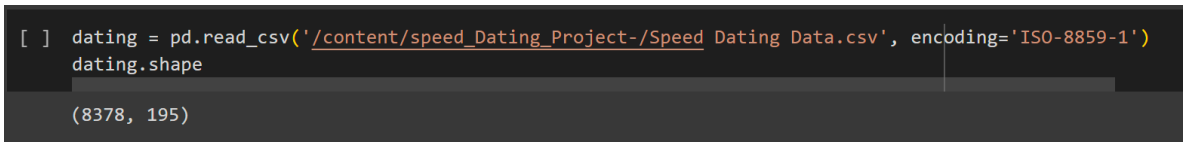

Figura 1 (Carga de datos)

Eliminación de Variables Redundantes

Durante el análisis de las variables, se identificaron columnas que se consideraron redundantes o irrelevantes para el objetivo del proyecto. Dos de estas columnas, "field" y "career", contenían cadenas de texto que se referían a la misma carrera o campo laboral de manera diferente, lo que dificulta el proceso de limpieza y categorización de los datos. Por lo tanto, se optó por eliminar estas columnas. Además, se excluyeron columnas como 'idg', 'condtn', 'wave', 'round', 'position', y 'positin1', ya que no aportan información valiosa.

Tratamiento de Datos Faltantes

El tratamiento de datos faltantes se ha revelado como un desafío importante, y es esencial para garantizar que nuestro modelo de Machine Learning produzca resultados confiables y valiosos para su plataforma.

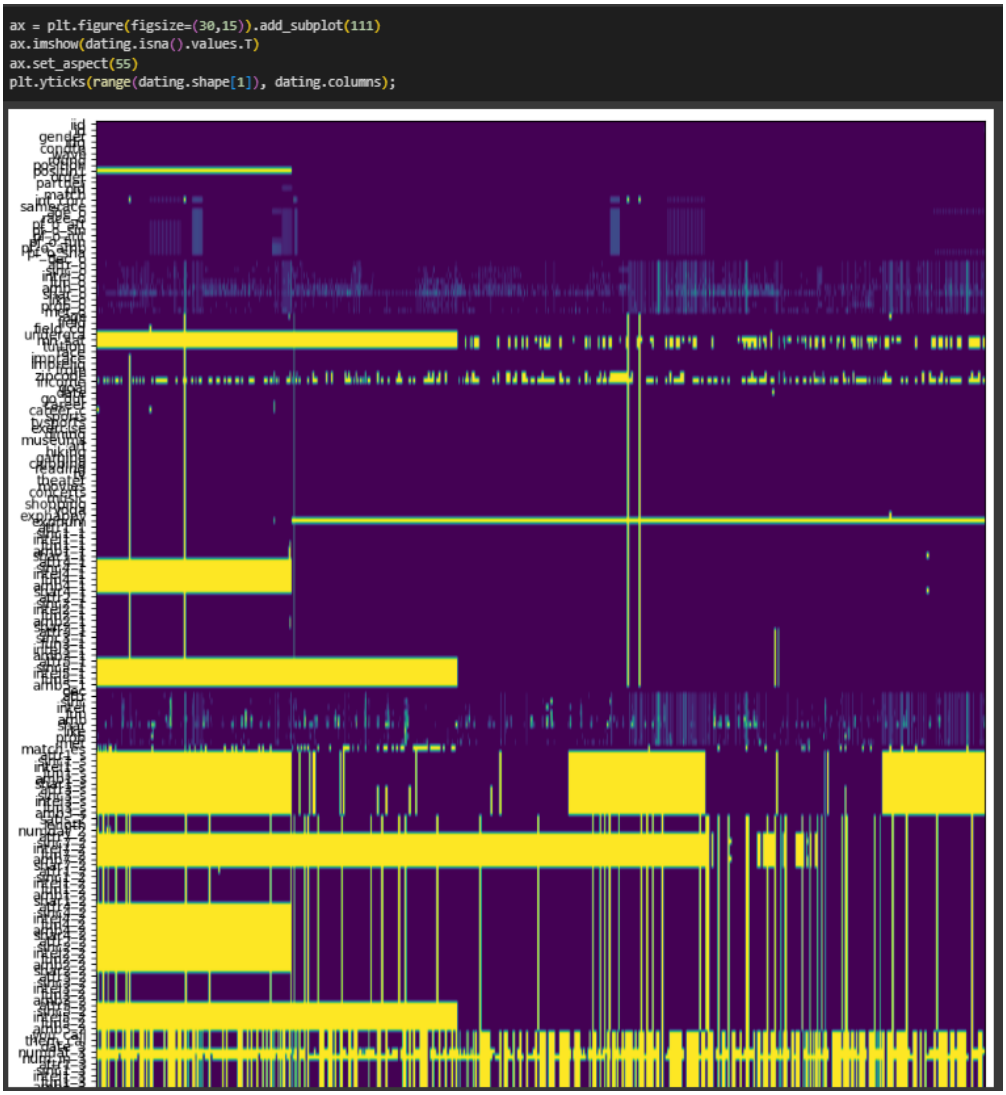

Figura 2 (Gráfico de datos faltantes)

Hemos implementado estrategias de manejo de datos faltantes que reflejan nuestro compromiso con la calidad de los resultados. Para las columnas con un alto porcentaje de datos faltantes, hemos optado por una estrategia de segmentación del dataset. Esta estrategia consiste en dividir el conjunto de datos en áreas más manejables, lo que nos permite enfocar nuestros esfuerzos en grupos de variables específicos. Esta segmentación ha simplificado el proceso de imputación y análisis de datos faltantes.

En el caso de columnas con datos faltantes más complejos, hemos aplicado estrategias de imputación avanzada basadas en modelos. Una de las técnicas que hemos utilizado es la imputación por regresión lineal. Inicialmente, probamos llenar los datos nulos con 0 y aplicar una regresión lineal para estimar los pesos de las variables. Sin embargo, tras un análisis más profundo, determinamos que esta estrategia no producía resultados satisfactorios. Esto refleja nuestra atención constante a la calidad de los datos y la precisión de nuestro modelo.

El análisis de correlaciones entre variables desempeñó un papel fundamental en la imputación y eliminación de datos faltantes. Las correlaciones fuertes entre variables nos permitieron identificar relaciones que simplificaron la limpieza y el preprocesamiento de datos de manera precisa.

```
[ ]  date = date.drop(['iid','pid','field', 'from', 'career',], axis=1)

[ ]  date = date.drop(['dec','dec_o',], axis=1)# estan relacionadas directamente con a variable objetivo y sesgan el resultado del experimento

[ ]  date['int_corr'] = date['int_corr'].fillna(0)
     date['career_c'] = date['career_c'].fillna(15)# 15 corresponde a otras carreras
     date[['race_o','race','imprace', 'imprelig','goal', 'date', 'go_out']] = date[['race_o','race','imprace', 'imprelig','goal', 'date', 'go_out']].fillna(date.mode())
     date = date.fillna(date.mean()) # Para todo lo demas usar la media
```

Figura 3 (Estrategias de limpieza de datos)

Normalización de Datos

Con el fin de reducir el esfuerzo para los modelos al momento del entrenamiento se normalizan los datos utilizando la técnica de mínimos y máximos.

```
for col in date.columns:
  date[col] = MinMaxScaler().fit_transform(np.array(date[col]).reshape(-1,1))
```

Figura 4 (Normalización de datos)

Análisis de Correlaciones

Un análisis crucial se centró en las correlaciones entre variables. Identificar las relaciones entre variables permitió la imputación y eliminación de parámetros de manera precisa. Las correlaciones fuertes entre variables indicaron que una podía ser inferida a partir de otra, lo que simplificó el proceso de limpieza y preprocesamiento de datos.

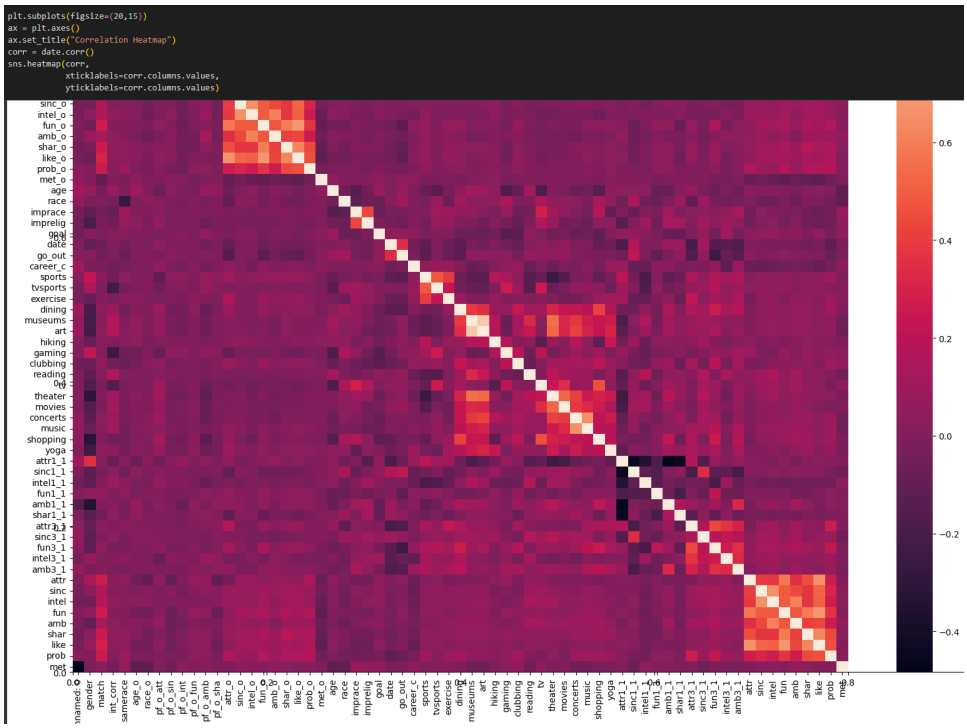

Figura 5 (Mapa de correlación)

Selección de Modelo de Machine Learning

Se implementaron varios modelos de Machine Learning para clasificar los datos en dos categorías: Si(1) y No(0), con el objetivo de predecir si una pareja volvería a verse. A continuación, se presenta un resumen del desempeño de los diferentes modelos:

Modelos Supervisados:

Implementación de modelos de clasificación, incluyendo DecisionTreeClassifier, LogisticRegression, SVC, SGDClassifier, RidgeClassifier y RandomForestClassifier.

```
models['DecisionTree'] = DecisionTreeClassifier(max_depth=3).fit(X,y)
models['LogisticRegression'] = LogisticRegression(max_iter = 500).fit(X,y)
models['SV'] = SVC(gamma=2, max_iter=300).fit(X,y)
models['SGD'] = SGDClassifier(loss="modified_huber", penalty="l2", max_iter=300).fit(X,y)
models['Ridge'] = RidgeClassifier(max_iter=300).fit(X,y)
models['RandomForest'] = RandomForestClassifier(n_estimators=2, max_depth=5).fit(X,y)
```

Figura 6 (Modelos supervisados)

División del dataset en conjuntos de entrenamiento y prueba.

Según los resultados, no se observó una diferencia significativa en el rendimiento entre estos modelos.

Modelos No Supervisados:

Exploración de técnicas de clustering para identificar patrones en los datos sin etiquetas, al utilizar 2 cluster para la tarea de clasificaciones se obtuvieron resultados deficientes, consideramos que esto se debe a la complejidad de los datos y cantidad de variables, lo que evita que el algoritmo converja hacia el resultado esperado. Se obtuvo un acierto del 50% clasificado los datos en un 50/50 para cada 1 de los 2 cluster, por lo tanto se concluye que el ruido entre los datos es elevado.

Implementación de algoritmos de agrupamiento como K-Means.

```
X = date.drop("match", axis=1)
y = date["match"]
```

```
km = KMeans(n_clusters=2, n_init='auto')
km.fit(X)
y_pred = km.predict(X)
y.shape
```

```
(8378,)
```

```
pd.Series(y_pred).value_counts() #Resultado de la clasificación
```

```
0    4190
1    4188
dtype: int64
```

```
round(accuracy_score(y, y_pred),4)#Porcentaje de acierto con respecto a la variable objetivo
```

```
0.5054
```

Figura 7 (Modelos no Supervisados)

Resultados y Métricas:

Realizamos las respectivas pruebas de las métricas de desempeño de los diferentes modelos que seleccionamos para nuestro proyecto. En cuanto a la precisión, obtuvimos resultados favorables, ya que casi todos los modelos mostraron una precisión superior al 80%. Esto significa que, en la mayoría de los casos, el modelo es capaz de predecir el problema planteado con alta exactitud. Estos resultados indican que los modelos iniciales tienen un buen potencial para ofrecer recomendaciones precisas a los usuarios de la plataforma de citas.

	DecisionTree	LogisticRegression	SV	SGD	Ridge	RandomForest
Accuracy	0.8409	0.8619	0.8927	0.8380	0.8564	0.8530
Recall	0.5027	0.2916	0.3340	0.0053	0.1347	0.1707
Specificity	0.9057	0.9711	0.9998	0.9976	0.9947	0.9837

Figura 8 (Métricas datos Train)

	DecisionTree	LogisticRegression	SV	SGD	Ridge	RandomForest
Accuracy	0.8254	0.8457	0.8210	0.8234	0.8449	0.8246
Recall	0.5034	0.2746	0.0183	0.0046	0.1396	0.0892
Specificity	0.8931	0.9658	0.9899	0.9957	0.9933	0.9793

Figura 9 (Métricas datos Test)

Sin embargo, al evaluar el recall, no se obtuvo una satisfacción tan grande. El recall más significativo que encontramos no supera el 40% Para la primera iteración. Esta limitación en el recall podría atribuirse a la necesidad de utilizar modelos de aprendizaje automático más complejos. Los modelos más complejos tienen una mayor capacidad para aprender patrones complejos en los datos, lo que puede llevar a una mejor identificación de las instancias positivas. Para abordar este desafío, se puede considerar la implementación de algoritmos más avanzados, como redes neuronales profundas, que son capaces de capturar relaciones más sutiles en los datos.

Otra estrategia para mejorar el recall es aumentar el tamaño del conjunto de datos. Esto se debe a que un conjunto de datos más grande es más probable que contenga instancias de todas las clases, incluidas las clases minoritarias. Un conjunto de datos más completo permitiría que el modelo tenga más ejemplos de casos positivos, lo que podría aumentar el recall. Por lo tanto, la adquisición y el uso de datos adicionales o la expansión del conjunto de datos actual pueden ser considerados como una forma de abordar el desafío del recall en el proyecto.

Estas consideraciones son cruciales para garantizar que los modelos sean capaces de proporcionar recomendaciones precisas y útiles a los usuarios, lo que a su vez mejora la experiencia en la plataforma de citas rápidas.

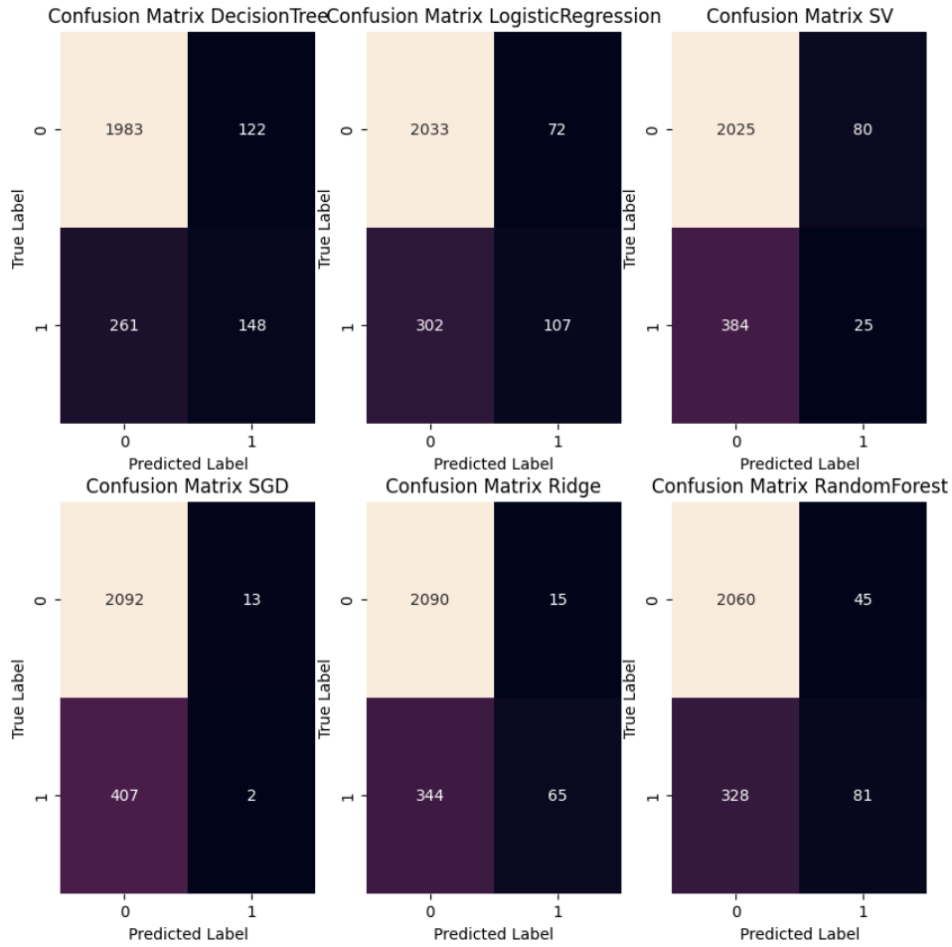

Figura 10 (Matriz de confusión)

La matriz de confusión es una herramienta que nos permite visualizar el desempeño de un modelo de clasificación mostrando el número de predicciones correctas e incorrectas. En este contexto de citas rápidas, la matriz de confusión nos ayuda a entender cómo los modelos están prediciendo si una pareja se volverá a ver.

En la matriz, los valores en la diagonal principal representan los aciertos, es decir, las instancias que fueron clasificadas correctamente. Los valores fuera de la diagonal principal representan los errores, que se dividen en dos tipos: falsos positivos (instancias clasificadas como "verdaderas" pero que son "falsas") y falsos negativos (instancias clasificadas como "falsas" pero que son "verdaderas").

Decision Tree

El modelo Decision Tree exhibe un rendimiento general bueno, con una precisión del 82.8%. Sin embargo, destaca por cometer un mayor número de falsos positivos en comparación con los otros dos modelos. Esto implica que es más propenso a clasificar una instancia como "verdadera" cuando en realidad es "falsa".

Logistic Regression

El modelo Logistic Regression muestra un rendimiento ligeramente inferior al Decision Tree, con una precisión del 81.2%. Sin embargo, tiene menos falsos positivos que el Decision Tree, lo que indica una menor probabilidad de clasificar una instancia como "verdadera" cuando es "falsa".

SVM (Support Vector Machine)

El modelo SVM presenta un rendimiento ligeramente inferior al Logistic Regression, con una precisión del 80.8%. Aunque su precisión general es menor, destaca por cometer menos falsos negativos en comparación con los otros modelos. Esto sugiere una menor probabilidad de clasificar una instancia como "falsa" cuando es "verdadera".

Despliegue e implementación del modelo

El despliegue de un proyecto de Machine Learning en el contexto de las citas presenta varios retos y consideraciones importantes. A continuación, se describen detalladamente algunos de los desafíos clave que deben abordarse al llevar a cabo la implementación en producción.

1. **Escalabilidad de la Infraestructura:** Uno de los desafíos iniciales es garantizar que la infraestructura de IT pueda manejar la carga de trabajo en producción. A medida que más usuarios interactúan con la plataforma, la capacidad de los servidores, la red y la base de datos debe ser escalable para mantener un rendimiento óptimo.

2. **Tiempo Real vs. Por Lotes:** Dependiendo de la aplicación, el despliegue puede requerir decisiones sobre si los resultados del modelo se generan en tiempo real o en lotes. Para una aplicación de citas rápidas, es crucial que las predicciones se realicen en tiempo real para ofrecer a los usuarios recomendaciones instantáneas. Esto implica la configuración de sistemas de procesamiento en tiempo real.

3. **Monitoreo y Mantenimiento Continuo:** Una vez en producción, es fundamental establecer un sistema de monitoreo constante para evaluar el rendimiento del modelo en tiempo real. Esto implica el seguimiento de métricas de calidad del modelo, como precisión y recall, para detectar cualquier degradación en el rendimiento. El mantenimiento continuo es necesario para reentrenar el modelo con nuevos datos y ajustar los hiperparámetros según sea necesario.

4. **Privacidad y Seguridad de los Datos:** La protección de la privacidad de los datos de los usuarios es una consideración crítica. Los datos utilizados para entrenar el modelo deben ser anonimizados y protegidos de posibles fugas de información. Además, se deben implementar medidas de seguridad robustas para proteger los datos y prevenir amenazas de ciberseguridad.
5. **Aprendizaje en Línea:** A medida que la plataforma recopila más datos de interacciones de los usuarios, es beneficioso explorar técnicas de aprendizaje en línea. Esto permite que el modelo se adapte continuamente a los cambios en las preferencias y comportamientos de los usuarios sin necesidad de un proceso de reentrenamiento completo.
6. **Cumplimiento Legal y Ético:** El proyecto debe cumplir con las regulaciones legales y éticas relacionadas con la recopilación y el uso de datos personales. Se deben establecer políticas y procedimientos claros para garantizar la conformidad con leyes de protección de datos, como el RGPD en Europa.
7. **Pruebas y Validación en Producción:** Antes de la implementación completa, se deben llevar a cabo pruebas exhaustivas en un entorno de producción simulado para garantizar que el modelo funcione correctamente y no cause problemas inesperados en la plataforma.

Métricas ML vs Métricas del negocio

En el planteamiento inicial se proyectó que el modelo buscaría aumentar la calidad de los emparejamiento realizados por la aplicación de citas a un 80%, luego de analizar los resultados del modelo tanto para los datos de entrenamiento como para los de pruebas, se encontró que la tasa de aciertos para parejas compatibles es de alrededor de un 20%, lo cual no es un desempeño aceptable para la implementación. Con el fin de no descartar el modelo se analizaron otras métricas como la Specificity, con la cual se encontró que los modelos entrenados tienen más de un 95% de acierto para identificar parejas que no son compatibles.

Dados estos resultados se considera cambiar el enfoque del proyecto en cuanto al objetivo del modelo, pero conservando la idea principal de mejorar las sugerencias que se realizan dentro de la aplicación, ya que si se descartan en más de un 95% las parejas no compatibles según los estudios previos se espera un aumento en la satisfacción de los usuarios superior al 12% en los instrumentos de satisfacción de los diferentes canales de distribución.

Conclusiones

El análisis de citas rápidas y la predicción de la probabilidad de que una pareja vuelva a verse son tareas desafiantes pero importantes para mejorar la experiencia de los usuarios en la aplicación de citas. El preprocesamiento de datos y la segmentación de variables jugaron un papel esencial en el manejo de datos faltantes y redundantes.

Aunque varios modelos de Machine Learning se implementaron para la clasificación, se observó que el rendimiento general fue bastante similar entre ellos. Es importante destacar que, aunque los modelos pueden proporcionar predicciones precisas, la calidad de los datos sigue siendo un factor clave en el éxito de cualquier proyecto de Machine Learning.

A Pesar de haber tenido un acierto aceptable, superior al 80 % preccioncones para cada modelo, el recall no superó el 40% tanto para los datos de entrenamiento como para los de prueba, esto está asociado a la limpieza de datos y al tamaño del dataset, consideramos que aún existen variables que pueden tener una alta correlación y puede afectar los resultados esperados, también, los casi 9000 registros del dataset pueden llegar a considerarse insuficientes para llegar a resultados más precisos.

A pesar de haber tenido un enfoque claro en el objetivo de los modelos, luego de contrastar los resultados finales con los esperados, se identificó que la composición del dataset no era la mejor para encontrar parejas que tuvieran afinidad, sin embargo al analizar los resultados a través de la matriz de confusión, se encontró un enfoque que arrojaba una mejora significativa en las métricas de los modelos, indicando que los modelos son mejores identificados parejas que no tienen afinidad.