

DATATHON 4TA EDICIÓN

Anticipando el futuro financiero con Hey Banco

Presentado por el equipo dinamita
conformado por Anna Galilea Restrepo
Martínez

Problemas detectados

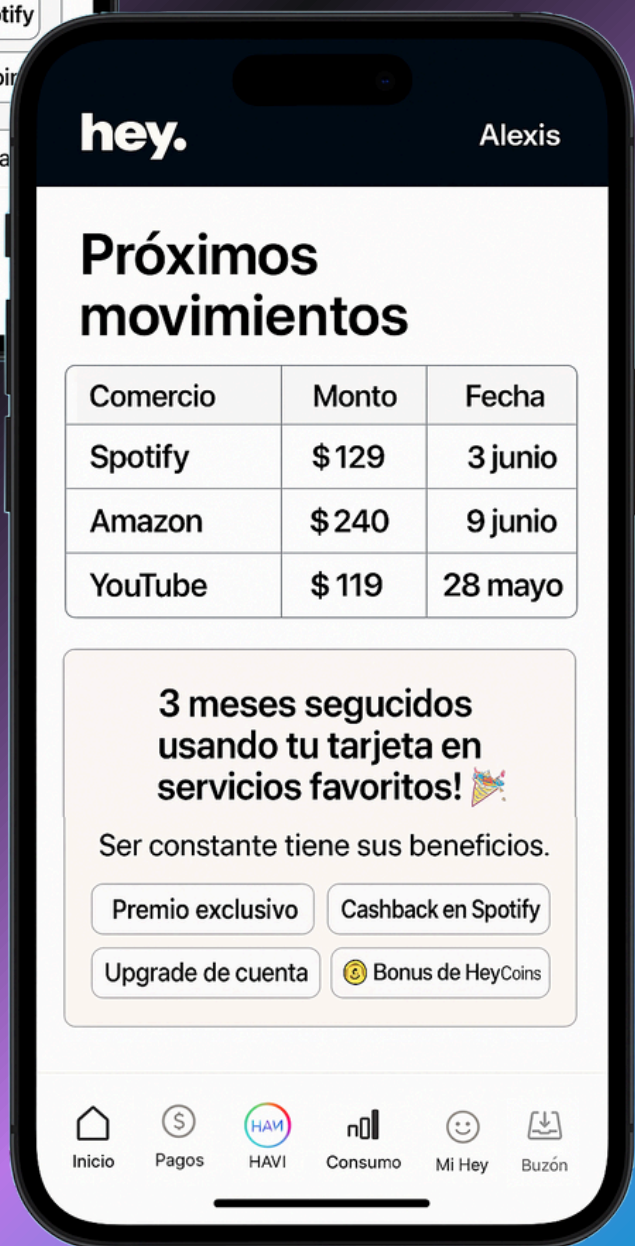
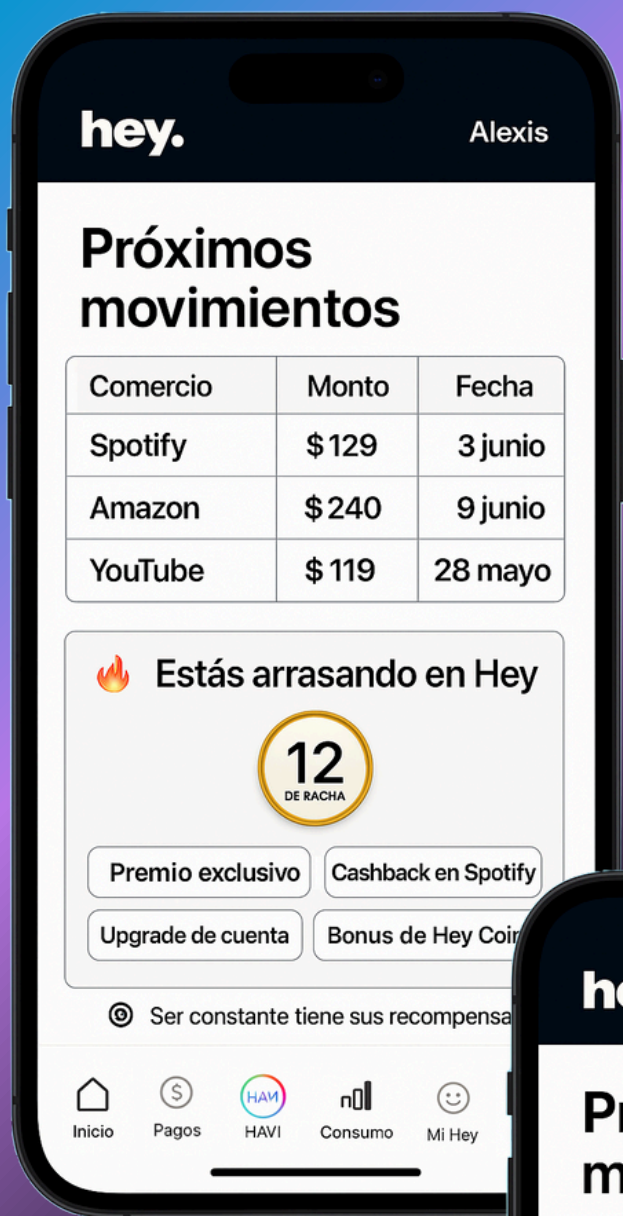
- 1 99% de los usuarios tienen patrones de gasto repetitivos, pero la app no los aprovecha.
- 2 No existe un espacio en la app que muestre próximos pagos estimados ni acción sugerida.
- 3 Esto limita la fidelización y el valor percibido.
- 4 No se aprovecha el historial transaccional para personalizar la experiencia del usuario.

Solución propuesta

- Módulo “Próximos movimientos” integrado al navbar.
- Predice:
 - Qué gasto hará el cliente
 - Cuándo lo hará
 - Cuánto gastará
- Se despliega como una tabla tipo dashboard con la información de: comercio, fecha estimada y monto esperado.
- También se puede implementar una racha o mecánicas de gamificación.
- Mejora la experiencia del usuario y promueve el uso de la tarjeta.

Además, según el uso frecuente del cliente con su tarjeta, se muestra una acción de negocio sugerida personalizada:

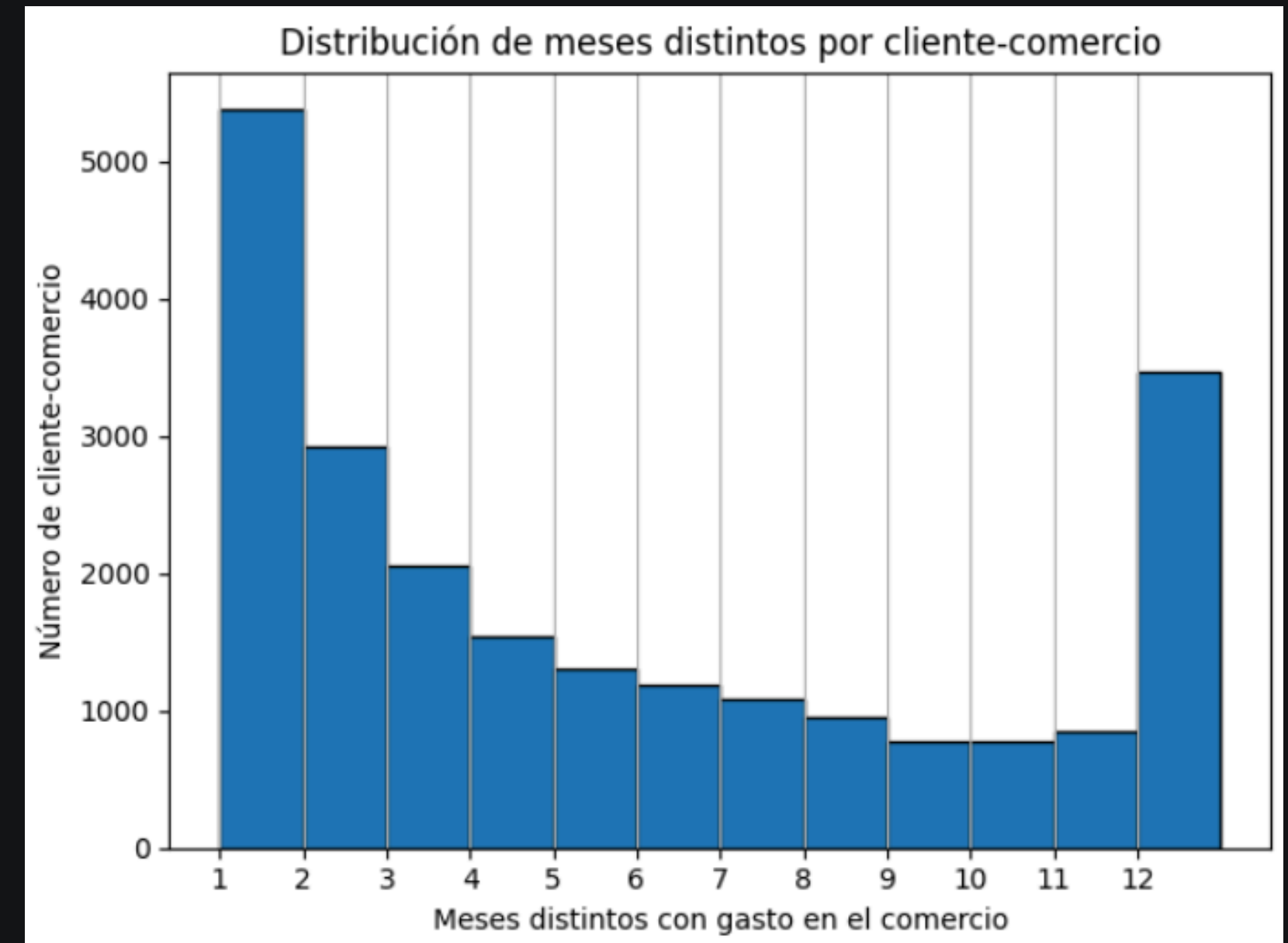
- Ultra constantes (12/12 meses): Premios exclusivos, upgrades de cuenta, cashback, lealtad
- Regularmente frecuentes (cada 2–3 meses): Automatización de pagos, promociones por recurrencia
- Gastos erráticos o casi nulos: Alertas personalizadas, campañas de retención, análisis de fuga



Cómo funciona

Para construir el módulo de “Próximos movimientos”, seguimos una metodología estructurada en distintas fases, combinando reglas, visualizaciones y modelos predictivos.

- Limpieza de datos
 - Unificación de las bases de datos
 - Variable año-mes
- Detección de recurrencias cliente-comercio
 - Conteo de meses distintos de incidencia por negocio
 - Decisión de la regla e histograma



- Clasificación de clientes acorde a su uso de la tarjeta
 - Regla para elegir si un cliente es recurrente
 - Histograma y decisión
- Modelos de predicción
 - Siguiendo comercio
 - Siguiendo monto por comercio
 - Siguiendo fecha por comercio

```

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import median_absolute_error

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=49)

rf = RandomForestRegressor(n_estimators=100, n_jobs=-1, random_state=49)
rf.fit(X_train, y_train)

preds = rf.predict(X_test)
mae = mean_absolute_error(y_test, preds)
medae = median_absolute_error(y_test, preds)
print(f"MAE mejorado: {mae:.2f} días")
print(f"Median Absolute Error: {medae:.2f}")
r2 = r2_score(y_test, preds)

print(f"R² del modelo: {r2:.4f}")

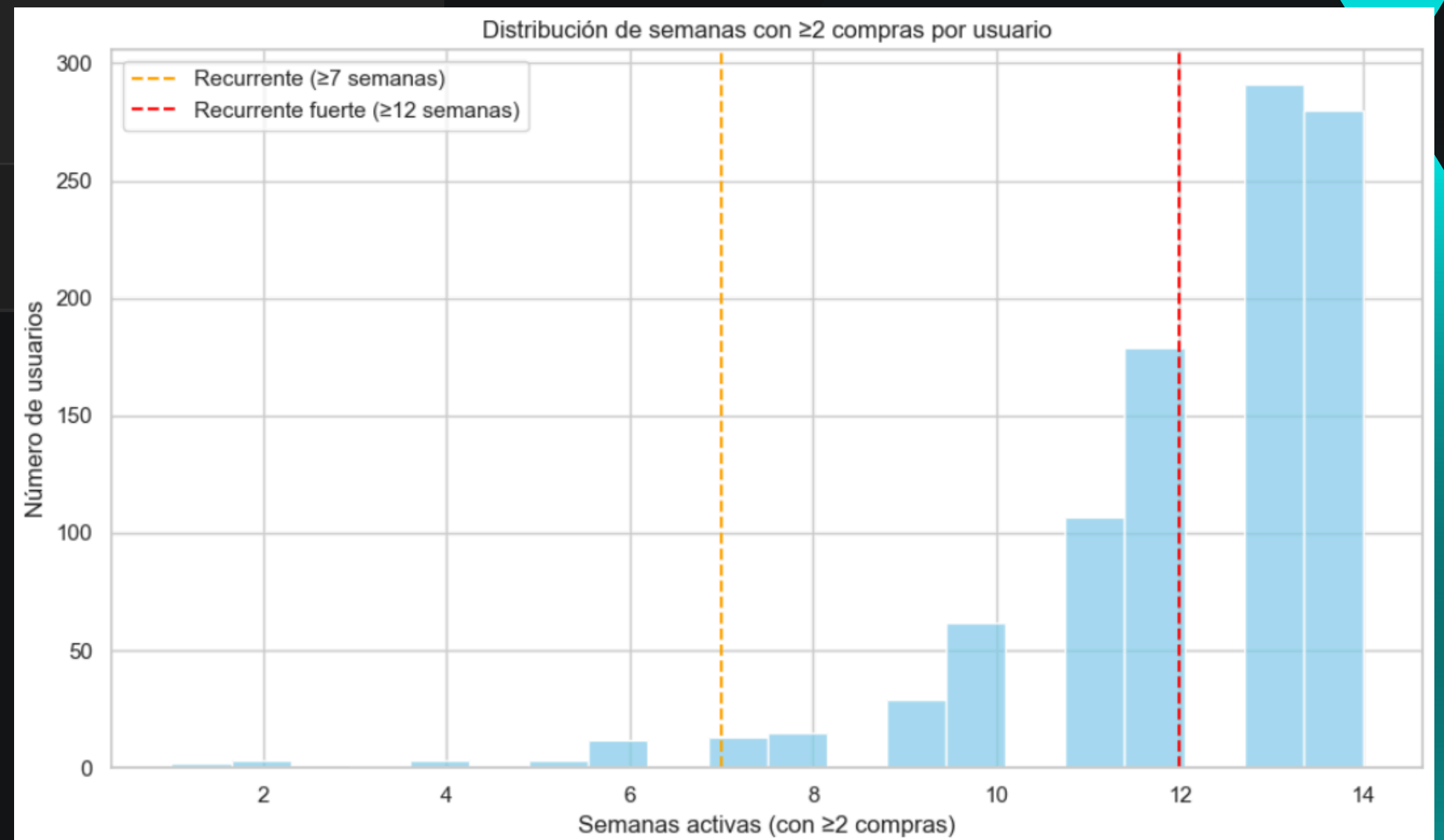
```

[9]

```

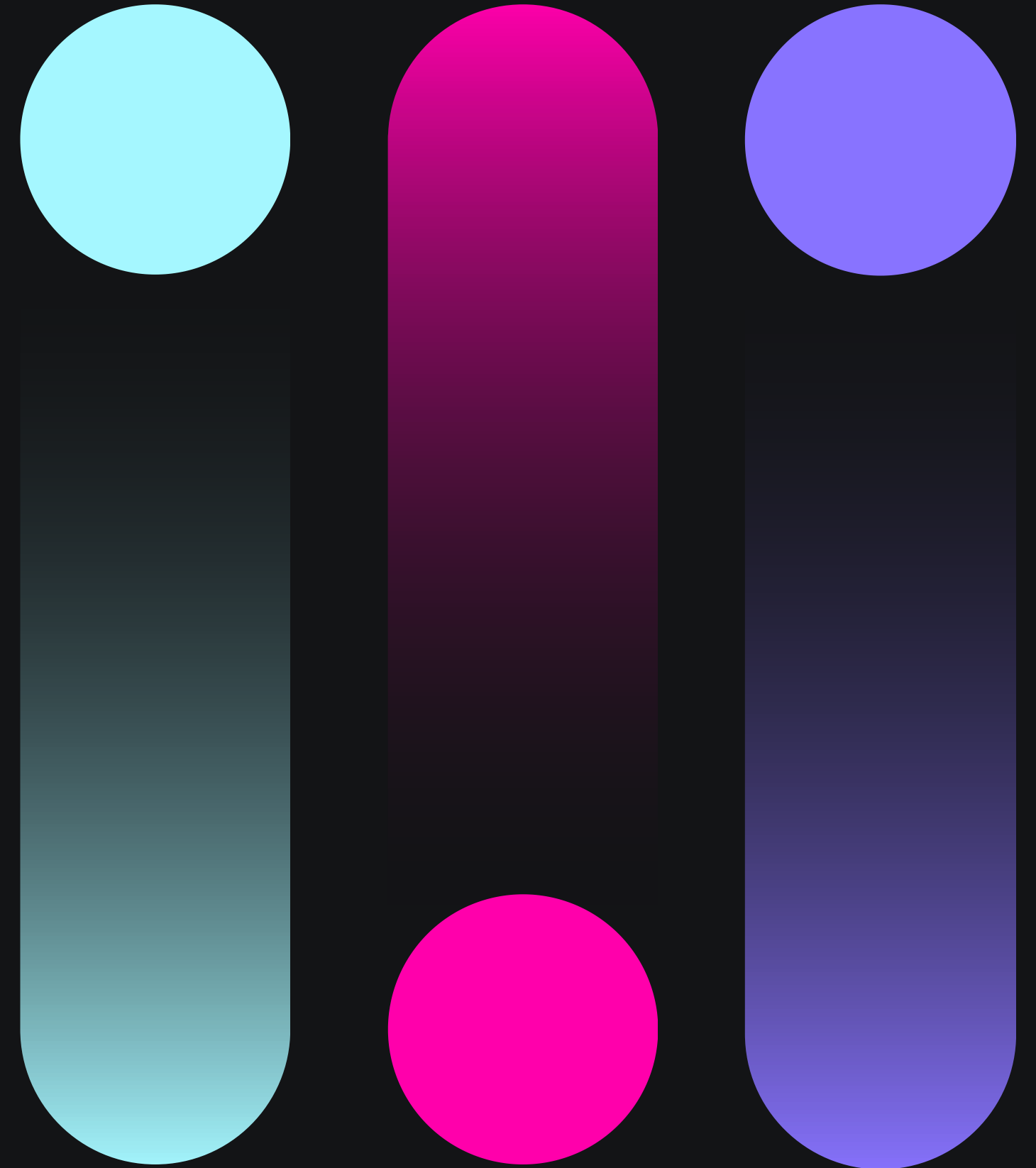
... MAE mejorado: 5.63 días
    Median Absolute Error: 2.43
    R² del modelo: 0.3631

```



Valor para el negocio

- Genera oportunidades de venta cruzada (Hey Pro, Hey Coins)
- Reduce el riesgo de fuga en perfiles inactivos
- Mejora el uso de la tarjeta como método de pago principal



Thank You

