

ثمانية thmanyah



توقع انسحاب المستخدمين

مع ابتسام علي

جدول المحتويات

ما هو انسحاب المستخدمين؟.....	3
لماذا يعد انسحاب العملاء مهمًا؟.....	3
أهمية التبؤ بالانسحاب باستخدام تعلم الآلة.....	4
للشخص التنفيذي.....	4
تعريف المشكلة تقنياً.....	4
فهم البيانات.....	5
نظرة عامة على البيانات.....	5
حجم البيانات.....	5
الأعمدة.....	6
مشاكل البيانات.....	7
التحليل الاستكشافي.....	7
الرسومات البيانية.....	8
الاستنتاج النهائي من التحليل الاستكشافي للبيانات.....	14
هندسة الخصائص.....	15
أهداف هندسة الخصائص.....	15
منهجية العمل.....	15
جدول الخصائص المستخرجة (Feature Engineering)	16
بناء النماذج.....	17
تجهيز البيانات للنموذج.....	17
نماذج التعلم الآلي المستخدمة.....	18
تقييم النماذج.....	18
مقاييس التقييم المستخدمة.....	18
تقدير Logistic Regression	19
تقدير Random Forest	19
تقدير XGBoost Classifier.....	20
بعد ضبط العتبة Logistic Regression 0.55	20
مقارنة نتائج النماذج.....	21
نشر النموذج وبناء واجهة برمجية.....	21
طريقة تشغيل المشروع.....	22
قيود وتحديات المشروع.....	24
تحسينات المستقبلية.....	25
الخاتمة	25

ما هو انسحاب المستخدمين؟

انسحاب العملاء Customer Attrition هو مقياس يعبر عن نسبة العملاء الذين توقفوا عن استخدام خدمة الشركة أو أنهوا تعاملهم معها خلال فترة زمنية محددة.

يتم حساب معدل الانسحاب عادة باستخدام العادلة التالية:

$$\text{معدل الانسحاب} = \frac{\text{عدد العملاء الذين غادروا خلال الفترة}}{\text{عدد العملاء في بداية الفترة}}$$

هذا المقياس يستخدم على نطاق واسع لتقدير استقرار قاعدة العملاء وفهم سلوكهم على المدى المتوسط والطويل.

لماذا يعد انسحاب العملاء مرئياً؟

يعد انسحاب العملاء من أهم المؤشرات الحيوية للأعمال، وذلك للأسباب التالية:

تكلفة الاحتفاظ أقل من تكلفة الاتساع في معظم القطاعات يكون الاحتفاظ بالعملاء الحاليين أقل تكلفة بكثير من جذب عملاء

جدد

اتساع عملاء جدد يتطلب:

- حملات تسويقية
- جهود مبيعات
- وقت لبناء الثقة

بينما العملاء الحاليون قد تم بالفعل تجاوز هذه المراحل معهم. إذا فالانسحاب يؤثر مباشرة على الإيرادات!

فقدان العملاء يعني:

- انخفاض الإيرادات المستقبلية
- انخفاض القيمة العمرية للعميل (Customer Lifetime Value)
- مؤشر على جودة المنتج أو الخدمة

ارتفاع معدل الانسحاب قد يشير إلى:

- تجربة مستخدم سيئة
- مشاكل في التسليم
- عدم تلبية توقعات العملاء
- منافسة أقوى في السوق

أهمية التنبؤ بالانسحاب باستخدام تعلم الآلة

تلعب نماذج تعلم الآلة دوراً محورياً في التنبؤ المبكر باحتمالية تسرب العملاء، مما يتيح للشركات اتخاذ إجراءات استباقية مثل تقديم عروض مخصصة أو تحسين تجربة المستخدم. بدلاً من رد الفعل بعد انسحاب العميل، يمكن للشركة التصرف قبل حدوث الانسحاب. يقدم هذا المشروع حلًا عمليًا ومتكملاً لهذه المشكلة، مع التركيز على الجوانب التحليلية والهندسية والإنتاجية.

الملخص التنفيذي

يهدف هذا المشروع إلى بناء نظام ذكي للتنبؤ بتسرب العملاء (Customer Churn) اعتماداً على بيانات سلوكيات تفصيلية تمثل تفاعل المستخدمين مع المنصة عبر الزمن. تم تنفيذ المشروع بأسلوب شامل يعطي دورة حياة تعلم الآلة كاملة، بدءاً من فهم البيانات الخام وتحليلها، مروراً بمعالجة البيانات وهندسة الخصائص، وصولاً إلى تدريب النموذج، وضبطه، وتقديره، ثم نشره كخدمة تنبؤية باستخدام واجهة برمجية (API) Docker وحوافيات. يركز الحل على التنبؤ بتسرب العملاء على مستوى المستخدم، وليس على مستوى الحدث، بما يتواافق مع التعريفات التجارية الواقعية لفهوم churn. النتيجة النهائية هي خدمة قابلة للتشغيل على أي بيئة، تتيح الحصول على احتمالية التسرب واتخاذ قرارات مبكرة مبنية على البيانات.

تعريف المشكلة تقنياً

صياغة المشكلة:

بناء نموذج تعلم آلة قادر على التنبؤ بما إذا كان المستخدم سيتسرب ($\text{Churn} = 1$) أو سيستمر في استخدام الخدمة ($\text{Churn} = 0$)، اعتماداً على سلوكه التاريخي.

نوع المشكلة:

تصنيف ثنائي (Binary Classification)، التعلم بإشراف (Supervised Learning)

مستوى التنبؤ:

مستوى المستخدم (User-level)، وليس مستوى الحدث (Event-level).

التحديات الأساسية:

- البيانات مبنية على الأحداث وليس على المستخدمين مباشرة.
- وجود عدم توازن واضح بين فئات churn و non-churn.
- الطبيعة الزمنية لسلوك المستخدم.
- احتمالية كبيرة لتسرب البيانات.

فهرم البيانات

نظرة عامة على البيانات

البيانات عبارة عن **سجل أحداث**: (Event Log) كل صف يمثل "حدث/تفاعل" قام به المستخدم داخل المنصة (مثل تشغيل أغنية، تسجيل دخول، إعجاب... إلخ).

هذا مهم لأنّه يعني أن البيانات ليست **ماشة كمنذجة على مستوى المستخدم**، بل تحتاج لاحقاً إلى تحويلها إلى خصائص مجمعة على مستوى المستخدم. (User-level Aggregation).

حجم البيانات:

- عدد السجلات: **543,705**
- عدد الأعمدة: **18**
- عدد المستخدمين (unique userId): **449**
- الفترة الزمنية التقريرية: من **2018-10-01** إلى **2018-12-01**

الأعمدة

تحتوي البيانات على 18 عمود بعضها تمثل أحداث والبعض الآخر متعلق بالمستخدم في الجدول التالي شرح مفصل لكل عمود:

الأهمية	ملاحظات جودة البيانات	عدد القيم المفقودة	نسبة القيم المفقودة	عدد القيم الفريدة	نوع البيانات	الوصف	العمود
عالية جداً	datetime يحتاج تحويل إلى وترتيب زمني	0	0.00%	513,108	int64	وقت الفعل (باليلي ثانية)	ts
عالية جداً	سيتم تجميع البيانات لاحقاً باستخدامه	0	0.00%	449	object	معرف المستخدم الفريد	userId
عالية	يُستخدم لحساب عدد الجلسات	0	0.00%	4,590	int64	معرف الجلسة	sessionId
عالية جداً	يحتوي أحداث churn؛ عدم توازن شديد لبعض القيم	0	0.00%	22	object	نوع الحدث/الصفحة	page
متوسطة	قد يفسر نقص بيانات المستخدم	0	0.00%	4	object	حالة المصادقة	auth
منخفضة	قيمة تحويلية محدودة	0	0.00%	2	object	نوع طلب HTTP	method
متوسطة	قد يعكس مشاكل تجربة المستخدم	0	0.00%	3	int64	كود استجابة HTTP	status
عالية جداً	free/paid؛ مهم جداً للتبؤ	0	0.00%	2	object	مستوى الاشتراك الحالي	level
متوسطة	يُستخدم لضبط الترتيب الزمني	0	0.00%	1,006	int64	ترتيب الحدث داخل الجلسة	itemInSession
متوسطة	كارديناليّ مرتفع؛ يحتاج تبسيط	15,700	2.89%	192	object	الموقع الجغرافي للمستخدم	location
عالية بعد المعالجة	نص طويل؛ غير مناسب مباشرة	15,700	2.89%	71	object	معلومات الجهاز والمتصفح	userAgent
منخفضة جداً	بيانات تعريفية؛ تستبعد	15,700	2.89%	275	object	اسم العائلة	lastName
منخفضة جداً	بيانات تعريفية؛ تستبعد	15,700	2.89%	345	object	الاسم الأول	firstName
عالية جداً	أساس حساب مدة البقاء (tenure)	15,700	2.89%	448	float64	وقت تسجيل المستخدم	registration
منخفضة- متوسطة		15,700	2.89%	2	object	جنس المستخدم	gender
منخفضة	مفقود طبيعياً لغير NextSong	110,828	20.38%	21,247	object	اسم الفنان	artist
منخفضة	كارديناليّ مرتفع جداً	110,828	20.38%	80,292	object	اسم الأغنية	song
عالية بعد التجميع	يُستخدم للتجميع (وقت الاستئماع)	110,828	20.38%	16,679	float64	مدة الأغنية بالثواني	length

مشاكل البيانات

القيم المفقودة

هناك نسبتين لقيم المفقودة في البيانات:

قيم مفقودة بنسبة ~2.89% في الأعمدة وهي:

location, userAgent, lastName, firstName, registration, gender
وجود قيم مفقودة في هذه الأعمدة طبيعي ولا يحتاج إلى معالجة حيث أنها تكون مفقودة في أحداث ك Guest , Logged out

قيم مفقودة بنسبة ~20.38% في الأعمدة:

artist, song, length

وهذا طبيعي لأن هذه الأعمدة لا تطلب هذه المعلومات.

الأعمدة عالية الكاردinاليق(High Cardinality)

بعض الأعمدة تحتوي على عدد كبير جدًا من القيم الفريدة مثل:

قيمة song (80,292)

قيمة artist (21,247)

قيمة location (192)

قيمة userAgent (71)

استخدام هذه الأعمدة مباشرة في النمذجة قد يؤدي إلى:

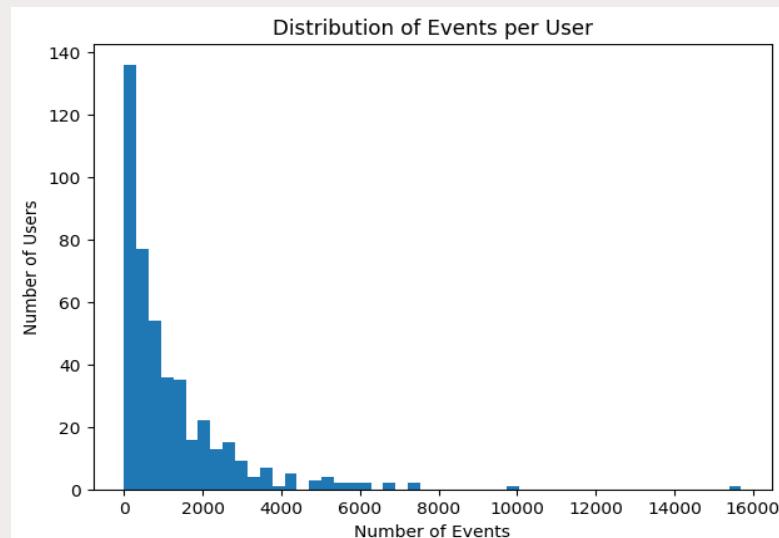
- تضخم عدد الخصائص
- Overfitting

التحليل الاستكشافي

يهدف التحليل الاستكشافي للبيانات إلى فهم سلوك المستخدمين داخل النظام قبل بناء نموذج التنبؤ بالانسحاب لفهم العلاقات بين الأعمدة المختلفة وتحليل سلوك المستخدم قدر الإمكان من خلال هذا التحليل، فمما بدراسة:

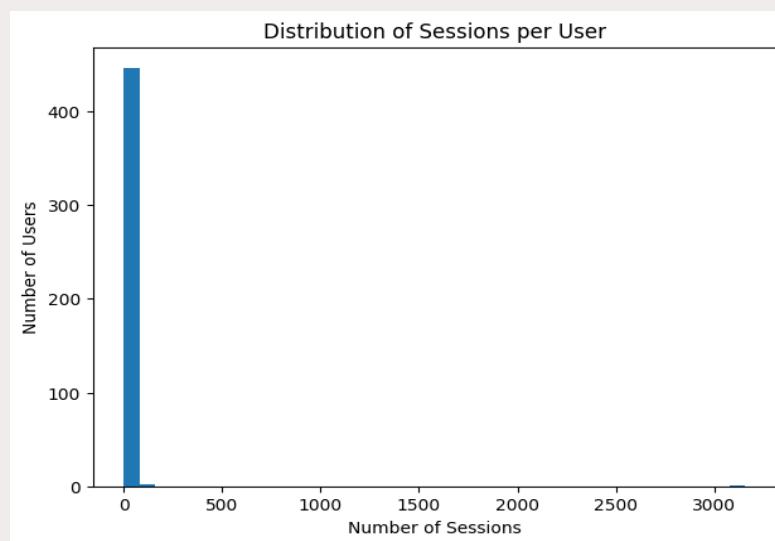
- حجم تفاعل المستخدمين (عدد الأحداث، عدد الجلسات)
- أنماط الاستخدام بمرور الوقت
- سلوك المستخدمين المنسحبين مقابل غير المنسحبين
- العلاقة بين مدة بقاء المستخدم (Tenure) واحتمالية الانسحاب
- الأحداث المرتبطة مباشرة بقرار الإلغاء

الرسومات البيانية



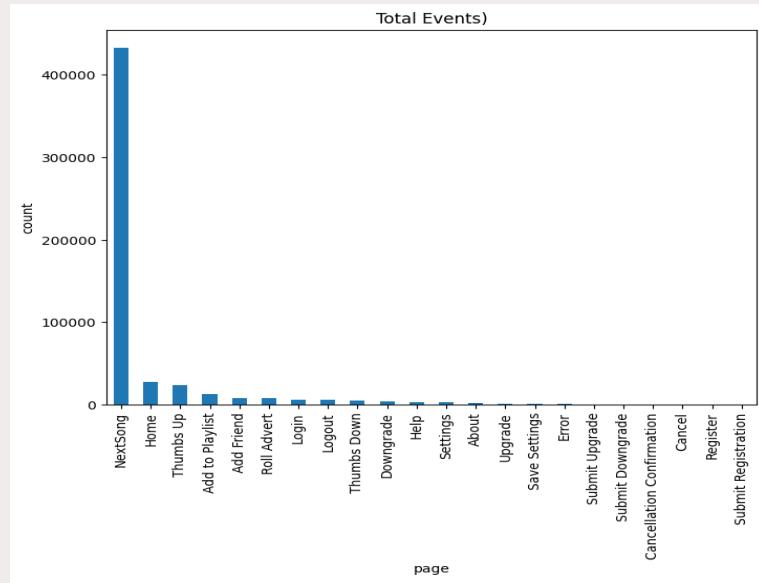
توزيع عدد الأحداث لكل مستخدم

يوضح هذا الرسم عدد التفاعلات (Events) التي قام بها كل مستخدم. أغلب المستخدمين لديهم عدد قليل من الأحداث ، في المقابل يوجد عدد صغير جداً من المستخدمين لديهم نشاط العالي. أيضاً البيانات منحرفة بشدة لليمين (Right-skewed) ، مما يشير إلى تفاوت كبير في استخدام المنصة



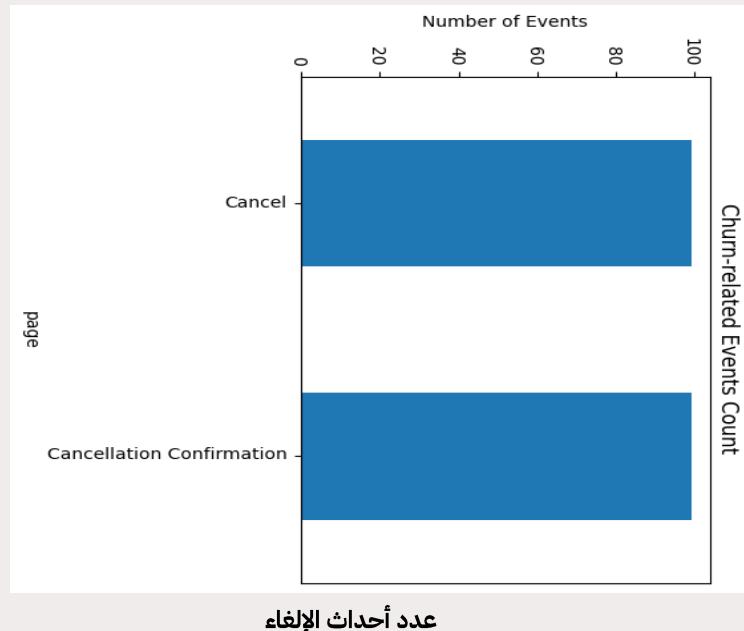
توزيع عدد الجلسات لكل مستخدم

يشير الرسم أن الغالبية العظمى من المستخدمين لديهم عدد جلسات منخفض قلة قليلة جداً لديها عدد جلسات مرتفع للغاية. أي أن **كثرة الجلسات ليست شائعة**.



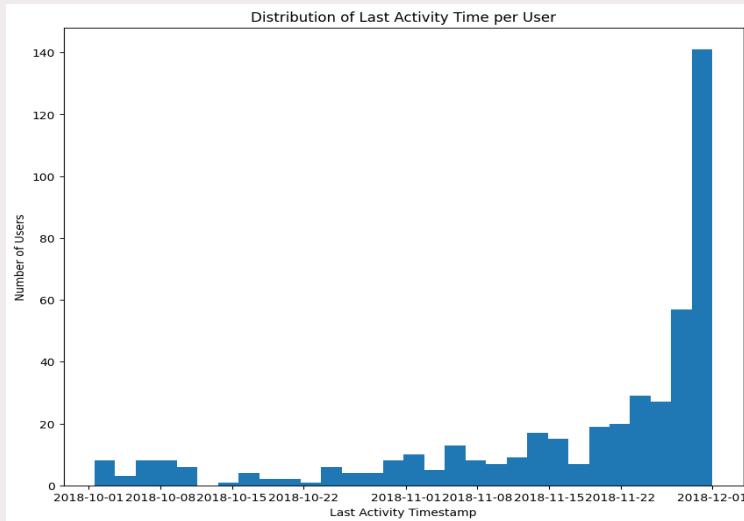
توزيع الأحداث حسب نوع الصفحة

صفحة NextSong هي الأكثر استخداماً بشكل واضح يليها الصفحات التي تمثل تفاعل إيجابي ك Thumbs up and Add to playlist. في المقابل أحداث الإلغاء (Cancel / Cancellation Confirmation) نادرة جداً، وهذا يوضح عدم توازن الفئات (Class Imbalance)، وهو عامل حاسم في تصميم النموذج والتقييم.



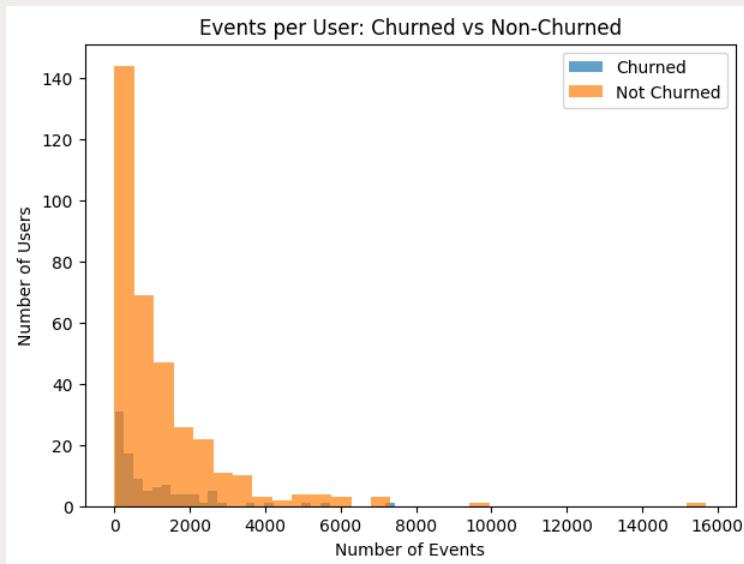
عدد أحداث الإلغاء

عدد أحداث الإلغاء منخفض جداً مقارنة بباقي الأحداث وهذا يؤكد أن قرار الإلغاء حدث نادر لكنه مهم. لذلك يجب تعريف الانسحاب على مستوى المستخدم وليس الحدث.



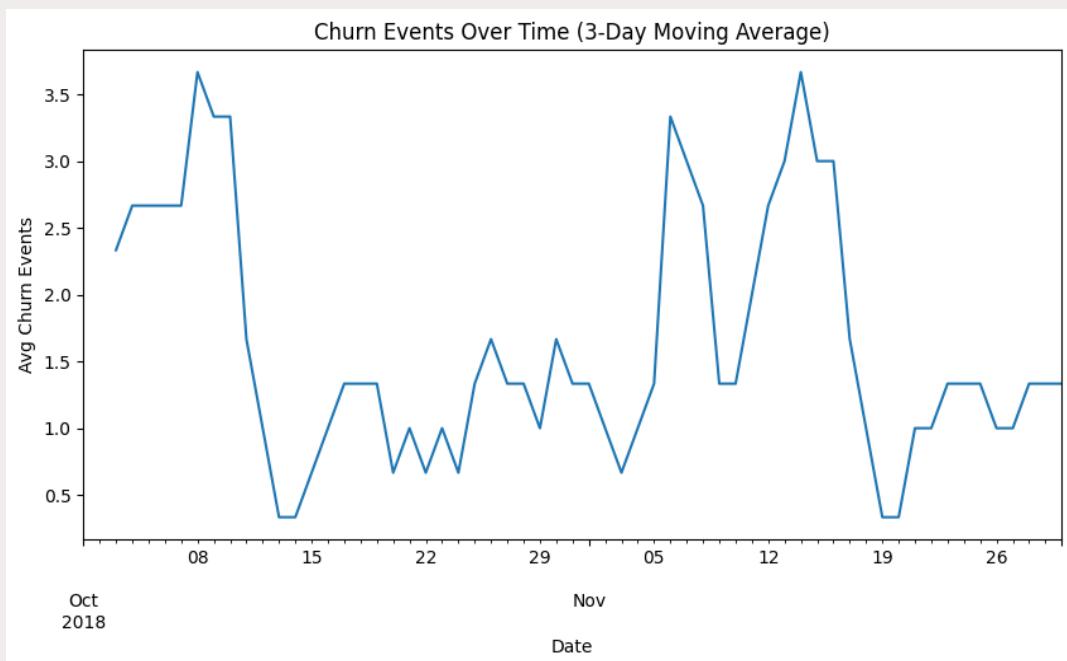
توزيع آخر نشاط لكل مستخدم بمرور الوقت

يوضح الرسم توقيت آخر نشاط للمستخدمين، ويبيّن أن معظم المستخدمين ظلوا نشطين حتى نهاية فترة الرصد البيانات، مما يدل على استمرارية تفاعلهم مع النظام، حقّ أئم كانوا أكثر نشاطاً نهاية الفترة . ويعود هذا المؤشر مهماً في تحليل الانسحاب، حيث أن المستخدمين ذوي النشاط القديم يكونون أكثر عرضة للانسحاب مقارنة بالمستخدمين ذوي النشاط الحديث.

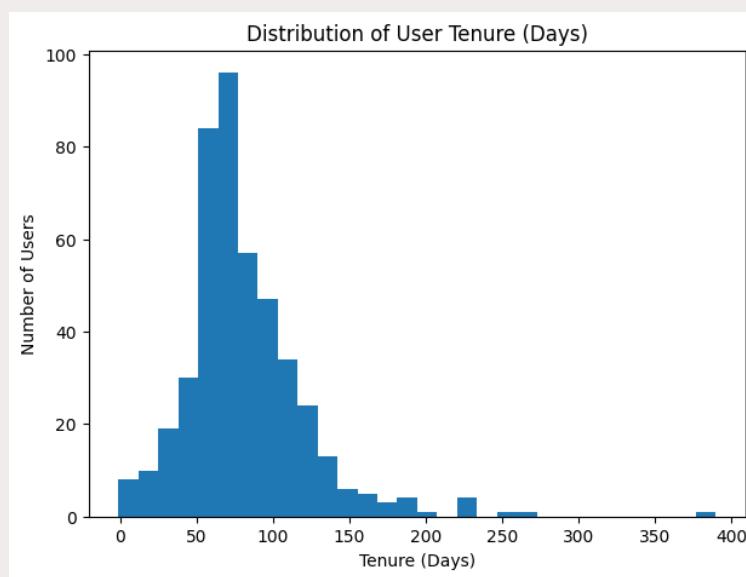


مقارنة عدد الأحداث بين المنسحبين وغير المنسحبين

يشير الرسم إلى أن المستخدمون غير المنسحبين لديهم عدد أحداث أعلى بكثير من المنسحبون حيث أن المنسحبون يظهرون نشاطاً أقل بشكل واضح. إنما فالنشاط المنخفض مؤشر قوي على الانسحاب.

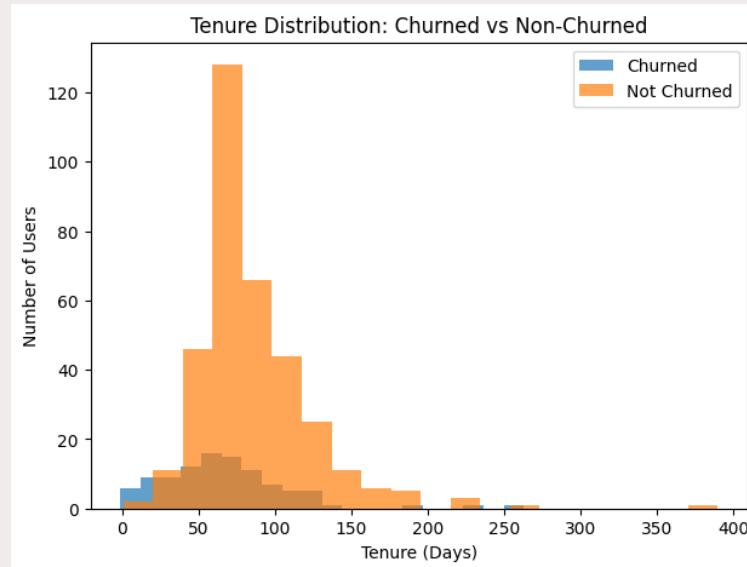


لا يوجد نمط زمني ثابت للانسحاب. تظهر قمم متفرقة تشير إلى أن الانسحاب قرار فردي وليس موسمياً.



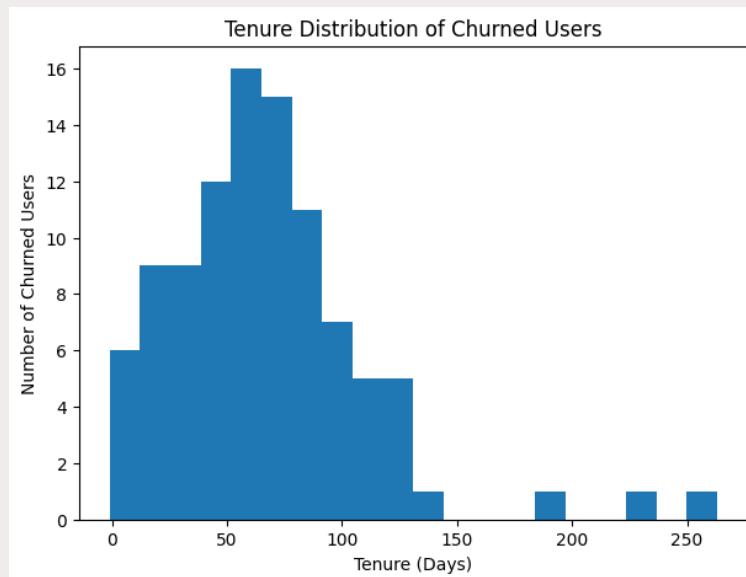
توزيع مدة بقاء المستخدم

يوضح هذا الرسم توزيع مدة بقاء المستخدمين، والتي تم احتسابها على أنها الفرق بين وقت تسجيل المستخدم وآخر نشاط له في البيانات. نلاحظ أن المستخدمين ذوي مدة البقاء القصيرة أكثر عرضة للانسحاب، بينما تشير مدة البقاء الأطول إلى مستوى أعلى من التفاعل والاستمرارية، مما يجعل هذا التغير من العوامل المهمة في التنبؤ بالإنسحاب.



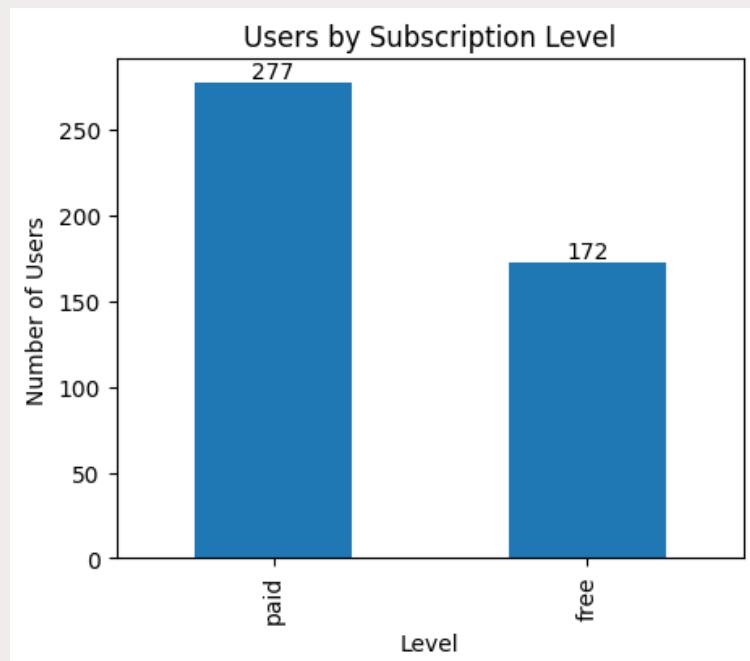
توزيع مدة بقاء المستخدم

المسحبون لديهم مدة بقاء أقصر بوضوح في المقابل غير المسحبين يمليون للبقاء لفترات أطول. نستنتج من هذا أن **Tenure** من **أقوى المؤشرات على الانسحاب**.



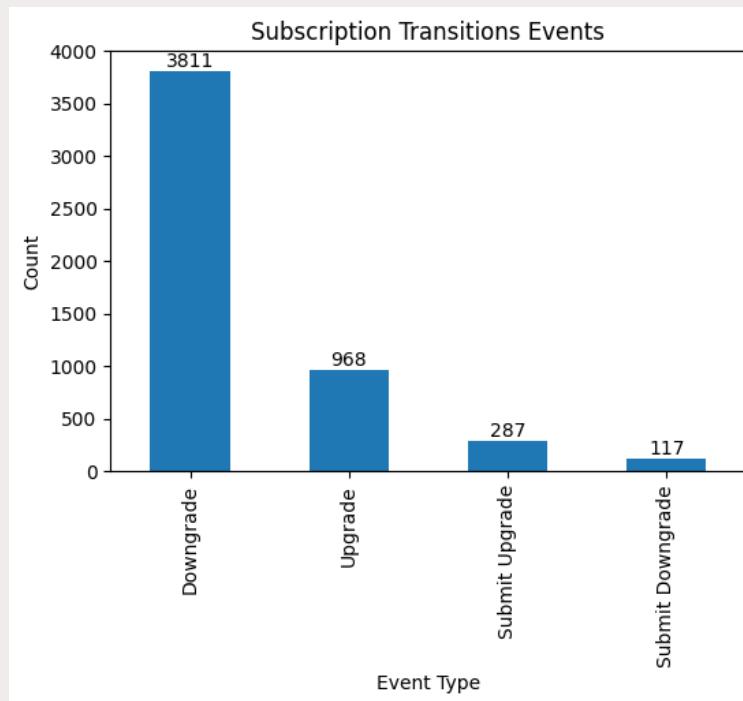
توزيع مدة بقاء المستخدمين للمسحبين

يمثل هذا الرسم توزيع مدة بقاء المستخدمين الذين قاموا بالانسحاب (Churned Users)، حيث يوضح عدد الأيام التي ظل فيها المستخدم نشطاً في النظام قبل أن ينسحب. الغالبية العظمى من المستخدمين المسحبين كانت مدة بقائهم أقل من **100 يوم**. يوجد عدد محدود جداً من المستخدمين الذين انسحبوا بعد فترات طويلة. يشير ذلك إلى أن المستخدمين الجدد هم الأكثر عرضة للانسحاب.



توزيع المستخدمين حسب مستوى الاشتراك

يبين الرسم أن عدد المستخدمين المدفوعين أعلى من المجانيين، مما يدل على وجود قيمة حقيقية في الخدمة، مع استمرار الحاجة لرراقبة سلوك الانسحاب لدى كلا الفئتين.



أحداث الانتقال بين مستويات الاشتراك

يظهر أن أحداث التخفيض (Downgrade) هي الأكثر تكراراً مقارنة بالترقية، مما يعكس احتمالية وجود عدم رضا لدى المستخدمين وبجعل هذا العامل مؤشراً مهماً للانسحاب.

الاستنتاج النهائي من التحليل الاستكشافي للبيانات

من خلال التحليل الاستكشافي للبيانات، تم التوصل إلى الاستنتاجات التالية:

- البيانات تعاني من عدم توازن واضح بين فئة المستخدمين النسجيين وغير النسجيين، مما يتطلب:
 - اختيار مقاييس تقييم مناسبة بدل الاعتماد على الدقة فقط.
 - استخدام (Threshold Tuning) لتحسين أداء النموذج.
- المستخدم النسجي يتميز بالخصائص التالية:
 - نشاط أقل داخل النظام.
 - عدد جلسات أقل مقارنة بالمستخدمين غير النسجيين.
 - مدة بقاء أقصر في النظام.
 - تفاعل محدود مع الليزات الاجتماعية مثل Thumbs Up و Add Friend و
- قرار الانسحاب:
 - يحدث على مستوى المستخدم وليس على مستوى الحدث الفردي.
 - لا يعتمد على حدث واحد فقط مثل Cancel Confirmation .
 - يتطلب تجميع سلوك المستخدم وتحليله عبر الزمن.
- نتائج التحليل الاستكشافي شكلت الأساس ل:
 - تصميم الخصائص التنبؤية(Feature Engineering)
 - اختيار نموذج قابل للتفسير لتسهيل فهم أسباب الانسحاب.
 - ضبط (Threshold Tuning) لتحقيق توازن أفضل بين مؤشرى Precision و Recall

هندسة الخصائص

بعد الانتهاء من التحليل الاستكشافي للبيانات وفهم سلوك المستخدمين والعوامل المرتبطة بالانسحاب، قمت بالبدء بـ **هندسة الخصائص** بهدف تحويل البيانات الخام إلى خصائص رقمية تعبر بشكل أفضل عن سلوك المستخدم وقابلة للاستخدام من قبل نماذج التعلم الآلي.

أهداف هندسة الخصائص

- تمثيل سلوك المستخدم على مستوى المستخدم بدلاً من مستوى الحدث.
- تلخيص التفاعل الزمني للمستخدم في مجموعة خصائص قابلة للتفسير.
- تعزيز قدرة النموذج على التمييز بين المستخدم المنسحب وغير المنسحب.
- تقليل الضوضاء والاعتماد على خصائص ذات دلالة سلوكية واضحة.

منهجية العمل

- تم تجميع البيانات على مستوى **userId** بدلاً من الاعتماد على السجلات الفردية.
- لكل مستخدم، تم استخراج ملخص شامل لسلوكه عبر فترة استخدامه للنظام.
- تم الاعتماد على نتائج EDA لاختيار الخصائص الأكثر ارتباطاً بالانسحاب.

جدول الخصائص المستخرجة(Feature Engineering)

اسم الخاصية	الشرح التفصيلي
num_events	يمثل إجمالي عدد التفاعلات التي قام بها المستخدم داخل النظام (مثل تشغيل الأغاني، التنقل بين الصفحات، التفاعلات المختلفة). يعكس هذا التغير مستوى النشاط العام للمستخدم، حيث أن المستخدمين المنسحبين غالباً ما يظهر لديهم عدد أقل من الأحداث.
num_sessions	عدد الجلسات الفريدة التي استخدم فيها المستخدم النظام. يدل هذا التغير على مدى تكرار عودة المستخدم إلى الخدمة، ويعود مؤشراً منها على الاستمرارية والارتباط بالخدمة.
events_per_session	متوسط عدد التفاعلات داخل الجلسة الواحدة، ويتم حسابه بقسمة عدد الأحداث على عدد الجلسات. يعكس عمق التفاعل داخل الجلسة، حيث قد يمتلك بعض المستخدمين جلسات قليلة لكنها غنية بالتفاعل.
num_songs	عدد الأغاني التي قام المستخدم بتشغيلها خلال فترة استخدامه للنظام. يعد مؤشراً مباشراً على استهلاك المحتوى الموسيقي واهتمام المستخدم بالخدمة الأساسية.
total_listen_time	إجمالي وقت الاستماع للمستخدم، ويعبر عن الزمن الكلي الذي قضاه في الاستماع للمحتوى. هذا التغير يعكس مستوى الارتباط بالخدمة بشكل أوضح من عدد الأغاني فقط.
Add Friend	عدد مرات استخدام المستخدم لـ زيز إضافة الأصدقاء. يعكس هذا التغير مستوى التفاعل الاجتماعي داخل النسخة، وقد أظهر التحليل أن انخفاض هذا النوع من التفاعل يرتبط بارتفاع احتمالية الانسحاب.
Thumbs Up	عدد مرات الإعجاب بالمحظى. يدل على رضا المستخدم عن المحتوى المقدم، ويعود مؤشراً إيجابياً على تجربة المستخدم.
Thumbs Down	عدد مرات عدم الإعجاب بالمحظى. يعكس احتمالية عدم الرضا، وقد يشير إلى تجربة استخدام سلبية في بعض الحالات.
Add to Playlist	عدد مرات إضافة الأغاني إلى قوائم التشغيل. يدل على نية المستخدم بالاحتفاظ بالمحظى والعودة إليه لاحقاً، وهو مؤشر على الاستمرارية.
last_level	آخر مستوى اشتراك للمستخدم (مجاني أو مدفوع) بناءً على آخر نشاط مسجل. يعبر عن الحالة النهائية للمستخدم قبل الاستمرار أو الانسحاب.
num_downgrades	عدد مرات تخفيض الاشتراك من مدفوع إلى مجاني. يعد من أقوى المؤشرات السلوكية المرتبطة بعدم الرضا واحتمالية الانسحاب.
tenure_days	مدة بقاء المستخدم في النظام، وتمثل الفرق بين تاريخ التسجيل وآخر نشاط له. من أكثر الخصائص ارتباطاً بالانسحاب، حيث يميل المستخدمون ذوو المدة الأقصر إلى الانسحاب بشكل أكبر.
device_type	نوع الجهاز المستخدم(Desktop / Mobile / Tablet)، وتم استخلاصه من حقل userAgent. قد يؤثر نوع الجهاز على تجربة الاستخدام وسلوك التفاعل.
churn	متغير الهدف (Label) الذي يحدد ما إذا كان المستخدم قد انسحب من الخدمة (1) أو لم ينسحب (0). تم تعريفه على مستوى المستخدم وليس الحدث الفردي.
churn_time	وقت حدوث الانسحاب في حال وجوده. يستخدم لأغراض تحليلية و زمنية فقط، وليس كمدخل للنموذج .

بناء النماذج

تجزيز البيانات للنموذجة

كانت الخطوة الأولى في مرحلة النماذجة هي فصل المتغيرات المستقلة عن المتغير الهدف:

X (Features): •

تمثل جميع الخصائص السلوكية والهندسية للمستخدم، باستثناء:

- churn ○
- churn_time. ○

وذلك لمنع أي تسرب بيانات(Data Leakage) ، حيث أن هذه الأعمدة تحتوي على معلومات مباشرة أو زمنية مرتبطة بقرار الانسحاب

y (Target): •

العمود churn، وهو متغير ثنائي:

- 1 → المستخدم منسحب ○
- 0 → المستخدم غير منسحب ○

ترميز المتغيرات ذات القيم غير الرقمية (Categorical Encoding)

نظرًا لأن النماذج لا يمكنها التعامل مباشرة مع القيم النصية، قمنا بترميز الخصائص الفئوية التالية:

- last_level •
- device_type. •

(Train / Test Split)

تم تقسيم البيانات إلى:

- 80% تدريب •
- 20% اختبار •

باستخدام الإعدادات التالية:

- test_size = 0.2 •
- random_state = 42 •
- stratify = y •

سبب استخدام stratify=y:

- لأن البيانات غير متوازنة (عدد المنسحبين أقل بكثير من غير المنسحبين) •
- يضمن هذا الخيار الحفاظ على نفس نسبة المنسحبين وغير المنسحبين في مجموعة التدريب والاختبار •
- يمكن تحيز النتائج ويعطي تقييماً أكثر واقعية لأداء النموذج •

نماذج التعلم الآلي المستخدمة

تمت تجربة عدة نماذج لتصنيف اختيار النموذج الأنسب للتبؤ بانسحاب المستخدمين . تم اختيار النماذج بناءً على شيوخ استخدامها، قابليتها للتفسير، وقدرتها على التعامل مع بيانات غير متوازنة.

جدول النماذج المستخدمة وخصائصها

القيود	أهم الخصائص	المزايا	نوع النموذج	النموذج
يفترض علاقة خطية	يعتمد على الاحتمالات. ينتج احتمال الانسحاب وليس فقط تصنيف ثلثائي	قابل للتفسير، مناسب لضبط العتبة، سريع	Linear / Probabilistic	Logistic Regression
أقل قابلية للتفسير	مجموعة أشجار قرار مستقلة	قوي ضد الضوضاء، يتعامل مع العلاقات غير الخطية	Ensemble (Tree-based)	Random Forest
حساس للإعدادات، قد يفرط في التعلم	تحسين تدريجي للأخطاء السابقة	قوي في البيانات المعقّدة	Boosting (Gradient Boosted Trees)	XGBoost Classifier

تقييم النماذج

نظرًا لأن بيانات الانسحاب غير متوازنة، فإن الاعتماد على الدقة (Accuracy) وحدتها غير كافية. لذلك تم استخدام عدة مقاييس تقييم لفهم أداء النماذج بشكل شامل.

مقاييس التقييم المستخدمة

• الدقة الإيجابية Precision

يوضح مدى صحة تنبؤات النموذج عندما يتوقع أن المستخدم منسحب، أي من بين جميع المستخدمين الذين صنفهم النموذج كمنسحبين، **كم مستخدماً كان منسحبًا فعليًا**

(False Positives)

• الاسترجاع - الأهم في هذا المشروع Recall

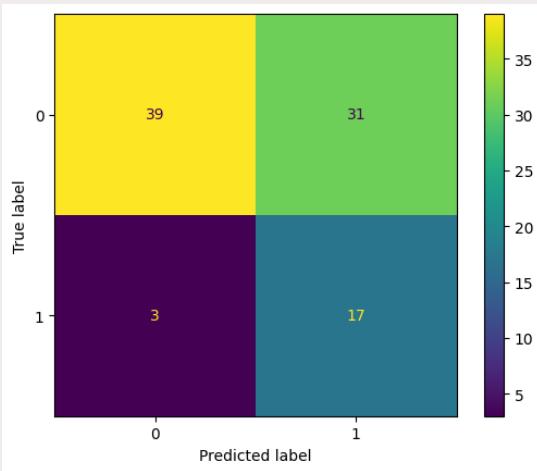
يوضح قدرة النموذج على اكتشاف المستخدمين النسحبين فعليًا، أي من بين جميع المستخدمين الذين انسحبوا بالفعل، **كم مستخدماً نجح النموذج في التعرف عليه**

(False Negatives)

• توازن بين Precision و F1-score

قدرة النموذج على التمييز بين الفئتين عبر جميع العتبات **ROC-AUC**

تقييم نموذج Logistic Regression



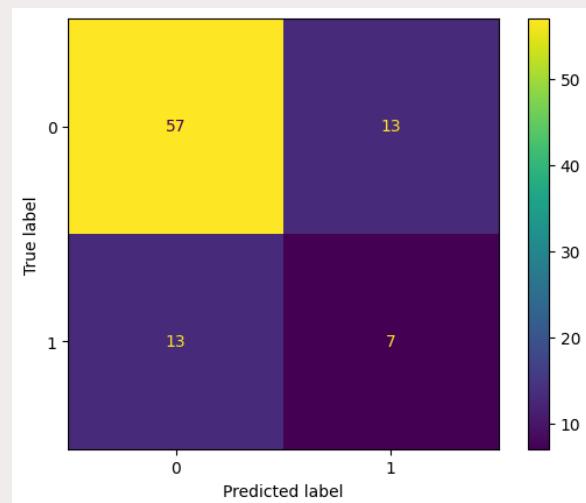
	precision	recall	f1-score	support
0	0.93	0.56	0.70	70
1	0.35	0.85	0.50	20
accuracy			0.62	90
macro avg	0.64	0.70	0.60	90
weighted avg	0.80	0.62	0.65	90

- النموذج نجح في اكتشاف معظم المستخدمين المنسحبين (Recall مرتفع)
- عدد حالات **False Negatives** منخفض جداً (ثلاث حالات فقط)، وهو أمر إيجابي.
- في المقابل، يوجد عدد ملحوظ من **False Positives**، مما أدى إلى انخفاض Precision.

الاستنتاج:

النموذج حساس لاكتشاف الانسحاب لكنه يطلق إنذارات خاطئة أكثر من اللازم، ليس بالضرورة أن يكون هذا أمر سليبي حيث أنه من الممكن أن يكون لهؤلاء المستخدمين (الإنذارات الخاطئة) سلوكيات تدل على أنهم في مرحلة ما سينسحبون فيجب الاهتمام بهم بشكل أكبر وإرسال العروض وما إلى ذلك.

تقييم نموذج Random Forest



	precision	recall	f1-score	support
0	0.81	0.81	0.81	70
1	0.35	0.35	0.35	20
accuracy			0.71	90
macro avg	0.58	0.58	0.58	90
weighted avg	0.71	0.71	0.71	90

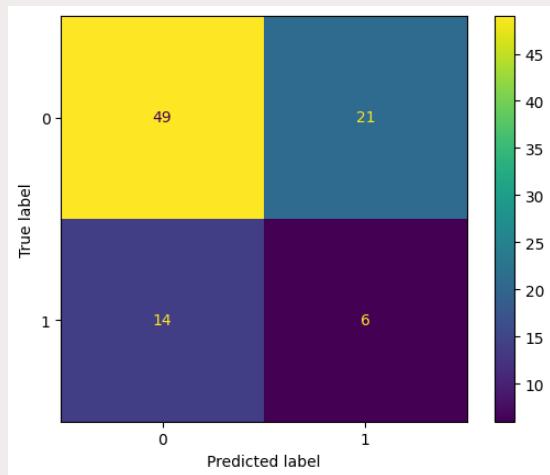
ROC-AUC: 0.5742857142857143

- النموذج يركز بشكل كبير على الفئة غير المنسحبة.
- عدد **False Negatives** مرتفع، أي أن النموذج فشل في اكتشاف عدد كبير من المستخدمين المنسحبين.
- رغم ارتفاع الدقة الكلية(Accuracy)، إلا أن الأداء على فئة Churn ضعيف.

الاستنتاج:

النموذج غير مناسب لحالة التنبؤ بالانسحاب، رغم دقته العامة الجيدة.

تقييم نموذج XGBoost Classifier



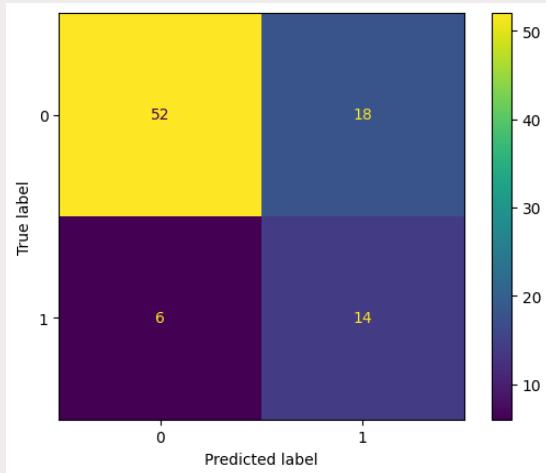
	precision	recall	f1-score	support
0	0.78	0.70	0.74	70
1	0.22	0.30	0.26	20
accuracy			0.61	90
macro avg	0.50	0.50	0.50	90
weighted avg	0.65	0.61	0.63	90
ROC-AUC:	0.5571428571428572			

- أظهر أداءً محدوداً في اكتشاف المستخدمين النسبيين.
- كل من Recall و Precision منخفضان.
- عدد كبير من حالات الانسحاب لم يتم اكتشافها.

الاستنتاج:

تعقيد النموذج لم ينعكس على أداء أفضل في اكتشاف الانسحاب.

تقييم نموذج Logistic Regression بعد ضبط العتبة 0.55



Chosen threshold: 0.5500000000000002				
	precision	recall	f1-score	support
0	0.90	0.74	0.81	70
1	0.44	0.70	0.54	20
accuracy			0.73	90
macro avg	0.67	0.72	0.68	90
weighted avg	0.79	0.73	0.75	90

- انخفاض عدد الإنذارات الخاطئة مقارنة بالنماذج الأساسية.
- لا يزال Recall مرتفعاً نسبياً، مع تحسن واضح في Precision.

الاستنتاج:

ضبط العتبة أدى إلى نموذج أكثر توازناً وقابلية للاستخدام العملي.

مقارنة نتائج النماذج

تم التركيز بشكل خاص على أداء الفئة المنسحبة ($\text{Churn} = 1$) نظرًا لأهمية اكتشاف حالات الانسحاب.

Precision (Churn=1)	Recall (Churn=1)	F1-score (Churn=1)	Accuracy	ROC-AUC	النموذج
35%	85%	50%	62%	78.4%	Logistic Regression (العتبة)
35%	35%	35%	71%	57.4%	Random Forest
22%	30%	26%	61%	55.7%	XGBoost Classifier
44%	70%	54%	73%	78.4%	Logistic Regression (العتبة 0.55) بعد ضبط

تم اختيار نموذج **Logistic Regression** بعد ضبط العتبة كنموذج نهائي، نظرًا لتحقيقه أفضل توازن بين مقاييس **Precision** و**Recall** مع الحفاظ على قدرة عالية على اكتشاف حالات الانسحاب. كما يتميز هذا النموذج ببساطته وقابليته للتفسير، مما يجعله الأقرب للاستخدام العملي واتخاذ القرارات.

نشر النموذج وبناء واجهة برمجية

تم تحويل النموذج النهائي إلى خدمة قابلة للاستخدام عبر واجهة برمجية (API) باستخدام **FastAPI**، مما يتيح إرسال بيانات للستخدم واستلام توقع الانسحاب بسهولة. ولضمان سهولة التشغيل والتوفيق بين البيئات المختلفة، تم تغليف المشروع باستخدام **Docker** ليكون جاهزًا للتشغيل على أي جهاز دون الحاجة لإعدادات محلية معقدة.

```
from fastapi import FastAPI
from pydantic import BaseModel, Field
import pandas as pd
import joblib
import json
from typing import Any, Dict

app = FastAPI(title="Churn Prediction API", version="1.0")

# Load artifacts
model = joblib.load("model.joblib")

with open("feature_columns.json", "r") as f:
    FEATURE_COLUMNS = json.load(f)

with open("threshold.json", "r") as f:
    THRESHOLD = float(json.load(f)["threshold"])

class UserFeatures(BaseModel):
    # numeric
    num_events: float = Field(..., ge=0)
    num_sessions: float = Field(..., ge=0)
    events_per_session: float = Field(..., ge=0)
    num_songs: float = Field(..., ge=0)
    total_listen_time: float = Field(..., ge=0)
    num_downgrades: float = Field(0, ge=0)
    tenure_days: float = Field(..., ge=0)

    # counts of actions (use API-friendly names)
    Add_Friend: float = Field(0, ge=0)
    Add_to_Playlist: float = Field(0, ge=0)
    Thumbs_Down: float = Field(0, ge=0)
    Thumbs_Up: float = Field(0, ge=0)

    # categoricals (raw, before one-hot)
    last_level: str = Field(..., examples=["free", "paid"])
    device_type: str = Field(..., examples=["Mobile", "Desktop", "Tablet", "Unknown"])

@app.get("/health")
def health() -> Dict[str, Any]:
    return {"status": "ok", "threshold": THRESHOLD}

@def build_model_input(payload: UserFeatures) -> pd.DataFrame:
    row = payload.model_dump()

    # map API keys to the original column names used before get_dummies
    rename_map = {
        "Add_Friend": "Add Friend",
        "Add_to_Playlist": "Add to Playlist",
        "Thumbs_Down": "Thumbs Down",
        "Thumbs_Up": "Thumbs Up",
    }
    row = {rename_map.get(k, k): v for k, v in row.items()}
```

```

df = pd.DataFrame([row])

# replicate training encoding
df = pd.get_dummies(df, columns=["last_level", "device_type"], drop_first=True)

# align to training features
df = df.reindex(columns=FEATURE_COLUMNS, fill_value=0)

return df


@app.post("/predict")
def predict(payload: UserFeatures) -> Dict[str, Any]:
    X = build_model_input(payload)

    proba = float(model.predict_proba(X)[:, 1][0])
    pred = int(proba >= THRESHOLD)

    return {
        "churn_probability": proba,
        "threshold": THRESHOLD,
        "churn_prediction": pred,
    }

```

طريقة تشغيل المشروع

في البداية، يجب تحميل مجلد `churn_service` والذي يحتوي على جميع الملفات الازمة لتشغيل المشروع، بما في ذلك النموذج المدرب، ملفات الإعداد، وواجهة API.

تشغيل المشروع باستخدام Docker

.1. فتح Command Prompt أو PowerShell ثم الانتقال إلى مجلد المشروع:

```
cd path/to/churn_service
```

.2. بناء صورة Docker يتم تنفيذ هذه الخطوة مرة واحدة فقط:

```
docker build -t churn-api .
```

.3. تشغيل الخدمة:

```
docker run -p 8000:8000 churn-api
```

.4. بعد التشغيل، يمكن الوصول إلى واجهة التوثيق التفاعلية لـ API عبر الرابط:

```
http://localhost:8000/docs
```

ومن خلال هذه الواجهة يمكن إرسال بيانات المستخدم واختبار التنبؤ بحالة الانسحاب.

5. بعد الانتقال للواجهة ستظهر هذه الصفحة ، لتجربة النموذج يجب الضغط على try it out

The screenshot shows the "Churn Prediction API" documentation page. It features a "default" section with two main endpoints: "GET /health" and "POST /predict". The "POST /predict" endpoint is highlighted with a red box around its "Try it out" button. Below the endpoint, there are sections for "Parameters" (No parameters), "Request body" (required), and "Example Value | Schema". The "Request body" section contains a JSON object with various user activity metrics and device type.

```
{
  "num_events": 0,
  "num_sessions": 0,
  "events_per_session": 0,
  "num_songs": 0,
  "total_listen_time": 0,
  "num_downgrades": 0,
  "tenure_days": 0,
  "Add_Friend": 0,
  "Add_to_Playlist": 0,
  "Thumbs_Down": 0,
  "Thumbs_Up": 0,
  "last_level": "free",
  "device_type": "Mobile"
}
```

6. تعبئة البيانات ، في المثال أسفله قمت بإدخال بيانات تمثل مستخدماً نشطاً، لديه تفاعل مرتفع ومدة بقاء أطول. نتيجة تنبأ النموذج أظهرت أنه مستخدم غير منسحب واحتمالية انسحابه ضعيفة جداً.

The screenshot shows the "Edit Value | Schema" section with a large JSON object representing user data. Below it, the "Response body" section displays the API's response, which includes the calculated "churn_probability" (0.00001114308825676473), "threshold" (0.5500000000000002), and "churn_prediction" (0). There are "Copy" and "Download" buttons next to the response body.

```
{
  "num_events": 1200,
  "num_sessions": 80,
  "events_per_session": 15,
  "num_songs": 950,
  "total_listen_time": 15000,
  "num_downgrades": 0,
  "tenure_days": 180,
  "Add_Friend": 12,
  "Add_to_Playlist": 25,
  "Thumbs_Down": 3,
  "Thumbs_Up": 4,
  "last_level": "paid",
  "device_type": "desktop"
}
```

```
{
  "churn_probability": 0.00001114308825676473,
  "threshold": 0.5500000000000002,
  "churn_prediction": 0
}
```

7. تعبيئة البيانات ، في المثال الثاني قمت بإدخال بيانات تمثل مستخدماً منخفض التفاعل، مدة بقائه قصيرة ونشاطه محدود. نتيجة تنبؤ النموذج أظهرت أنه مستخدم منسحب واحتمالية انسحابه عالية جداً.

```

Edit Value Schema
{
  "num_events": 45,
  "num_sessions": 4,
  "events_per_session": 3,
  "num_songs": 20,
  "total_listen_time": 300,
  "num_downgrades": 5,
  "total_downgrade": 12,
  "Add_Friend": 0,
  "Add_to_Playlist": 0,
  "Thumbs_Down": 12,
  "Thumbs_Up": 0,
  "last_level": "free",
  "device_type": "mobile"
}

Response body
{
  "churn_probability": 0.6580985388622414,
  "threshold": 0.5500000000000002,
  "churn_prediction": 1
}

```

قيود وتحديات المشروع

على الرغم من النتائج الجيدة التي حققها النموذج، إلا أنني واجهت الكثير من التحديات والقيود من أهمها:

- **محدودية الوقت المتاح للمشروع**
كون الوقت محدد (وأنا أيضاً أعمل بوظيفة ذات دوام كامل مما قلل الوقت أكثر) لم يكن بالإمكان إجراء تجارب أوسع على النماذج أو تطبيق تحسينات إضافية كان من الممكن أن تؤدي إلى أداء أفضل للنموذج.
- **عدم توازن البيانات (Class Imbalance)**
عدد المستخدمين غير النسحيين أكبر بكثير من المستخدمين النسحيين، مما يؤثر على دقة التنبؤ بفئة الانسحاب ويطلب تقنيات معالجة متقدمة.
- **حجم البيانات المستخدم محدود نسبياً**
بعد أن تم تجميع البيانات على مستوى المستخدم أصبح حجمها محدود مما قيد قدرة النموذج على التعلم من أنماط أكثر تنوعاً للسلوك طويلاً المدى.
- **الاعتماد على تجميع السلوك**
حيث لم يتم استخدام نماذج زمنية تأخذ تسلسل الأحداث بشكل مباشر في الاعتبار.
- **غياب التحقق من الأداء في بيئة إنتاج حقيقة**
إذ لم يتم تطبيق مراقبة مستمرة لأداء النموذج بعد النشر.(Model Monitoring)

التحسينات المستقبلية

يمكن تطوير هذا المشروع مستقبلاً وتحسين نتائجه من خلال:

- **تجربة نماذج أكثر تقدماً**
 - مثل النماذج الزمنية (LSTM / RNN) التي تستطيع تحليل تسلسل سلوك المستخدم بمرور الوقت.
- **تحسين التعامل مع عدم توازن البيانات**
 - باستخدام تقنيات مثل:
 - SMOTE
 - Class Weighting
 - Cost-sensitive learning
- **توسيع عملية ضبط العوامل (Hyperparameter Tuning)**
 - لإيجاد الإعدادات المثلث لككل نموذج وتحقيق أداء أعلى.
- **إضافة نظام مراقبة للنموذج**
 - لتابعه:
 - تدهور الأداء
 - تغيير سلوك المستخدمين (Data Drift & Concept Drift)
- **ربط النموذج بواجهة أمامية أو نظام حقيقى**
 - لاستخدام التنبؤات في اتخاذ قرارات عملية مثل حملات الاحتفاظ بالمستخدمين.

الخاتمة

تم في هذا المشروع بناء نظام متكامل للتنبؤ بانسحاب المستخدمين اعتماداً على تحليل سلوكهم داخل المنصة. شمل العمل فهم البيانات وتحليلها استكشافياً، ثم تصميم ميزات تمثل سلوك المستخدم عبر الزمن مع تجنب تسرب البيانات.

تمت تجربة عدة نماذج تعلم آلي وتقييمها باستخدام مقاييس مناسبة لطبيعة عدم توازن البيانات، مع التركيز على اكتشاف حالات الانسحاب. وبعد ضبط العتبة، تم اختيار نموذج **Logistic Regression** كنموذج نهائي لتحقيق أفضل توازن بين الدقة والاستدعاة.

أخيراً، تم نشر النموذج باستخدام **FastAPI** وتغليفه عبر **Docker** ليكون جاهزاً للاستخدام والتشغيل في بيئات مختلفة، مما يجعل الحل قابلاً للتطبيق العملي والتوسّع مستقبلاً.