ثمانية thmanyah

توقع انسحاب المستخدمين

مع ابتسام علي

جدول المحتويات

ما هو انسحاب المستخدمين؟

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

يتم حساب معدل الانسحاب عادة باستخدام المعادلة التالية:

معدل الانسحاب = عدد العملاء الذين غادروا خلال الفترة ÷ عدد العملاء في بداية الفترة

هذا المقياس يُستخدم على نطاق واسع لتقييم استقرار قاعدة العملاء وفهم سلوكهم على المدى المتوسط والطويل.

لماذا يُعد انسحاب العملاء مهمًا؟

يُعد انسحاب العملاء من أهم المؤشرات الحيوية للأعمال، وذلك للأسباب التالية:
تكلفة الاحتفاظ أقل من تكلفة الاكتساب في معظم القطاعات يكون الاحتفاظ بالعملاء الحاليين أقل تكلفة بكثير من جذب عملاء جدد
اكتساب 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 التسعير
- عدم تلبية توقعات العملاء
- منافسة أقوى في السوق

أهمية التنبؤ بالانسحاب باستخدام تعلم الآلة

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

الملخص التنفيذي

يهدف هذا المشروع إلى بناء نظام ذكي للتنبؤ بتسرب العملاء (Customer Churn) اعتمادا على بيانات سلوكية تفصيلية تمثل تفاعل المستخدمين مع المنصة عبر الزمن.
تم تنفيذ المشروع بأسلوب شامل يغطي دورة حياة تعلم الآلة كاملة، بدءًا من فهم البيانات الخام وتحليلها، مرورًا بمعالجة البيانات وهندسة الخصائص ، وصولا إلى تدريب النموذج، و تقييمه، وضبطه، ثم نشره كخدمة تنبؤية باستخدام واجهة برمجية (API) وحاويات.Docker
يركز الحل على التنبؤ بتسرب العملاء على مستوى المستخدم، وليس على مستوى الحدث، بما يتوافق مع التعريفات التجارية الواقعية لمفهوم.churn
النتيجة النهائية هي خدمة قابلة للتشغيل على أي بيئة، تتيح الحصول على احتمالية التسرب واتخاذ قرارات مبكرة مبنية على البيانات.

تعريف المشكلة تقنيا

صياغة المشكلة:
بناء نموذج تعلم آلة قادر على التنبؤ بما إذا كان المستخدم سيتسرب (Churn = 1) أو سيستمر في استخدام الخدمة(Churn = 0) ، اعتمادا 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.

فهم 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 لأنه يعني أن البيانات **ليست جاهزة مباشرة كنمذجة على مستوى المستخدم**، بل نحتاج لاحقا إلى تحويلها إلى خصائص مجمّعة على مستوى المستخدم.(User-level Aggregation)

حجم البيانات:

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

الأعمدة

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

العمود	الوصف	نوع البيانات	عدد القيم الفريدة	نسبة القيم المفقودة	عدد القيم المفقودة	ملاحظات جودة البيانات	الأهمية
ts	وقت الفعل (بالميلي ثانية)	int64	513,108	0.00%	0	يحتاج تحويل إلى datetime وترتيب زمني	**عالية جدا**
userId	معرف المستخدم الفريد	object	449	0.00%	0	سيتم تجميع البيانات لاحقا باستخدامه	**عالية جدا**
sessionId	معرف الجلسة	int64	4,590	0.00%	0	يُستخدم لحساب عدد الجلسات	**عالية**
page	نوع الحدث/الصفحة	object	22	0.00%	0	يحتوي أحداثchurn ؛ عدم توازن شديد لبعض القيم	**عالية جدا**
auth	حالة المصادقة	object	4	0.00%	0	قد يفسر نقص بيانات المستخدم	**متوسطة**
method	نوع طلبHTTP	object	2	0.00%	0	قيمة تحليلية محدودة	**منخفضة**
status	كود استجابةHTTP	int64	3	0.00%	0	404قد يعكس مشاكل تجربة المستخدم	**متوسطة**
level	مستوى الاشتراك الحالي	object	2	0.00%	0	free/paid؛ مهم جدا للتنبؤ	**عالية جدا**
itemInSession	ترتيب الحدث داخل الجلسة	int64	1,006	0.00%	0	يُستخدم لضبط الترتيب الزمني	**متوسطة**
location	الموقع الجغرافي للمستخدم	object	192	2.89%	15,700	كارديناليتي مرتفع؛ يحتاج تبسيط	**متوسطة**
userAgent	معلومات الجهاز والمتصفح	object	71	2.89%	15,700	نص طويل؛ غير مناسب مباشرة	**عالية بعد المعالجة**
lastName	اسم العائلة	object	275	2.89%	15,700	بيانات تعريفية؛ تستبعد	**منخفضة جدا**
firstName	الاسم الأول	object	345	2.89%	15,700	بيانات تعريفية؛ تستبعد	**منخفضة جدا**
registration	وقت تسجيل المستخدم	float64	448	2.89%	15,700	أساس حساب مدة البقاء (tenure)	**عالية جدا**
gender	جنس المستخدم	object	2	2.89%	15,700		**منخفضة–متوسطة**
artist	اسم الفنان	object	21,247	20.38%	110,828	مفقود طبيعيًا لغير NextSong	**منخفضة**
song	اسم الأغنية	object	80,292	20.38%	110,828	كارديناليتي مرتفع جدا	**منخفضة**
length	مدة الأغنية بالثواني	float64	16,679	20.38%	110,828	يُستخدم للتجميع (وقت الاستماع)	**عالية بعد التجميع**

مشاكل البيانات

القيم المفقودة

هناك نسبتين للقيم 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 تملأ إلا عند حدث NextSong، بينما بقية الأحداث لا تتطلب هذه المعلومات.

الأعمدة عالية الكارديناليتي(High Cardinality)

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

song (80,292 قيمة)

artist (21,247 قيمة)

location (192 قيمة)

userAgent (71 قيمة)

استخدام هذه الأعمدة مباشرة في النمذجة قد يؤدي إلى:

- تضخم عدد الخصائص
- Overfitting

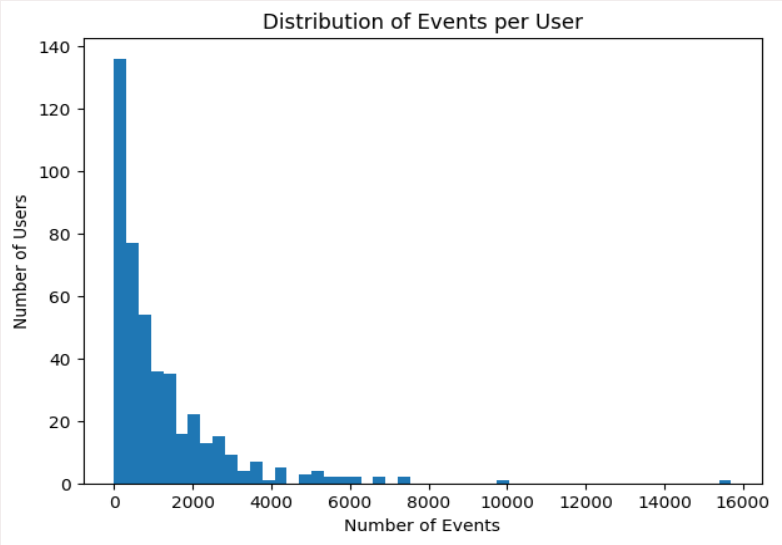
التحليل الاستكشافي

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، قمنا بدراسة:

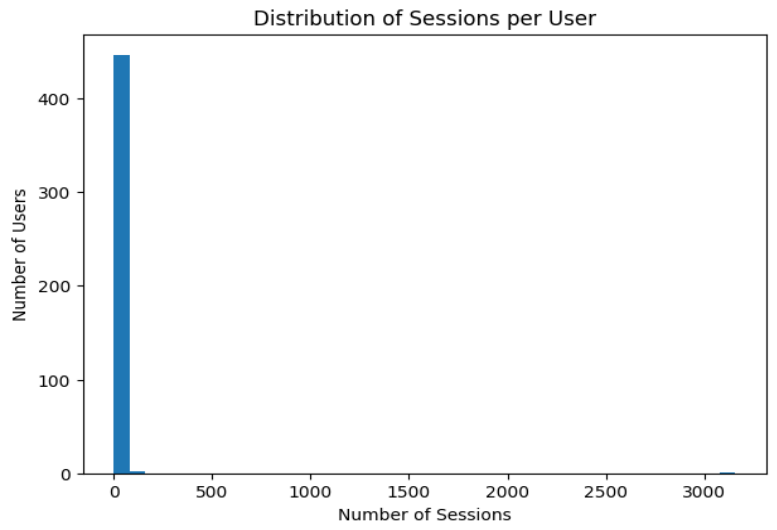
- **حجم تفاعل المستخدمين** (عدد الأحداث، عدد الجلسات)
- **أنماط الاستخدام بمرور الوقت**
- **سلوك المستخدمين المنسحبين مقابل غير المنسحبين**
- **العلاقة بين مدة بقاء المستخدم (Tenure) واحتمالية الانسحاب**
- **الأحداث المرتبطة مباشرة بقرار الإلغاء**

الرسومات البيانية



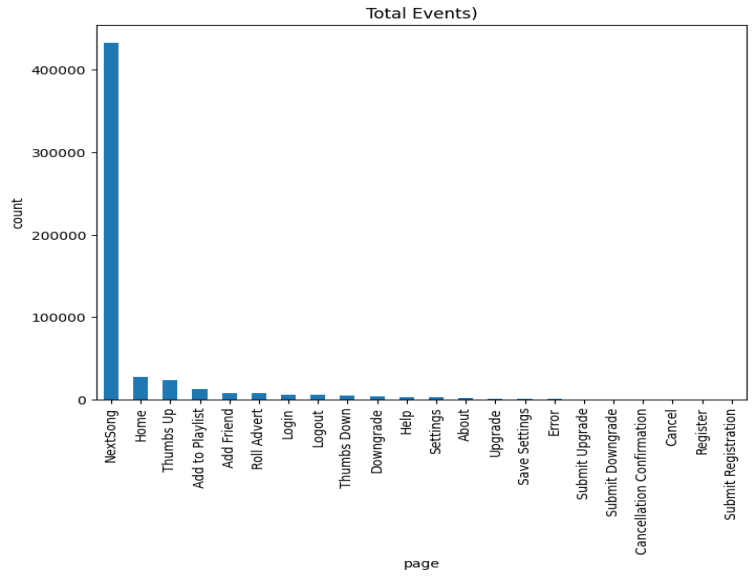
توزيع عدد الأحداث لكل

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



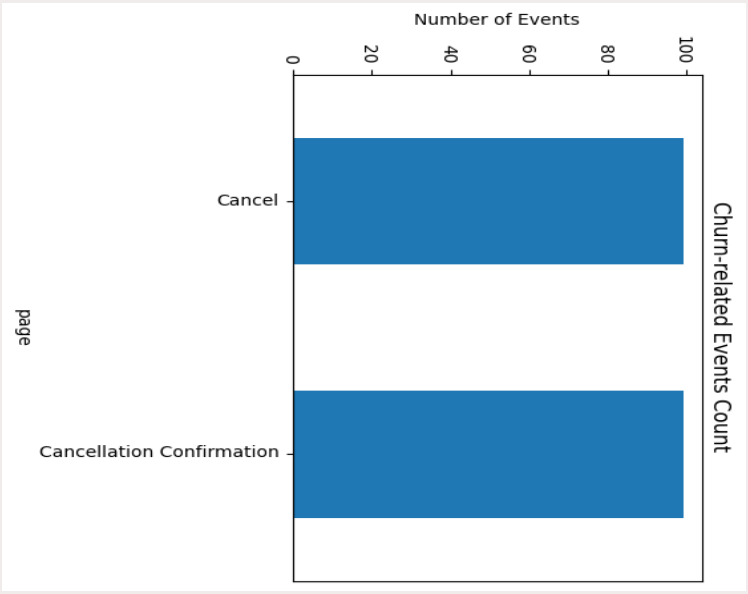
توزيع عدد الجلسات لكل مستخدم

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



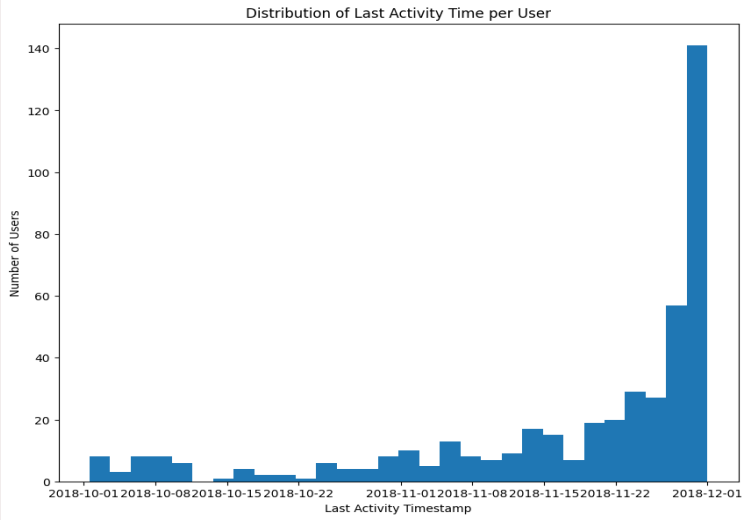
توزيع الأحداث حسب نوع الصفحة

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



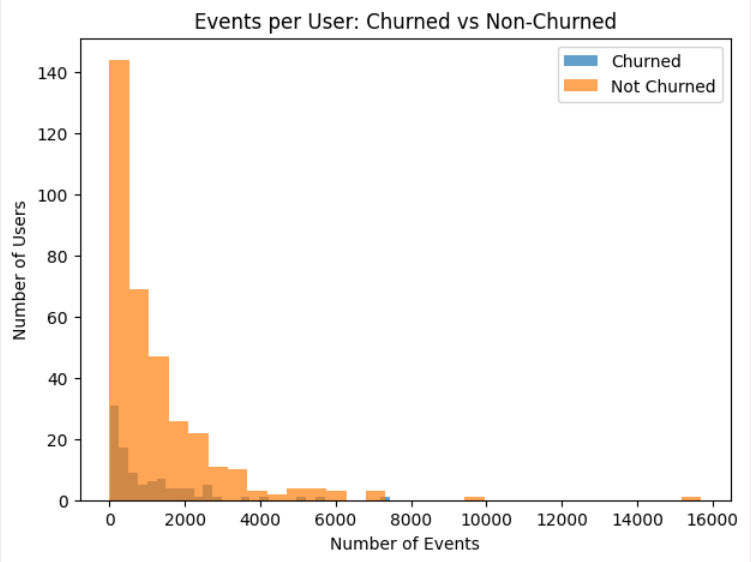
عدد أحداث الإلغاء

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



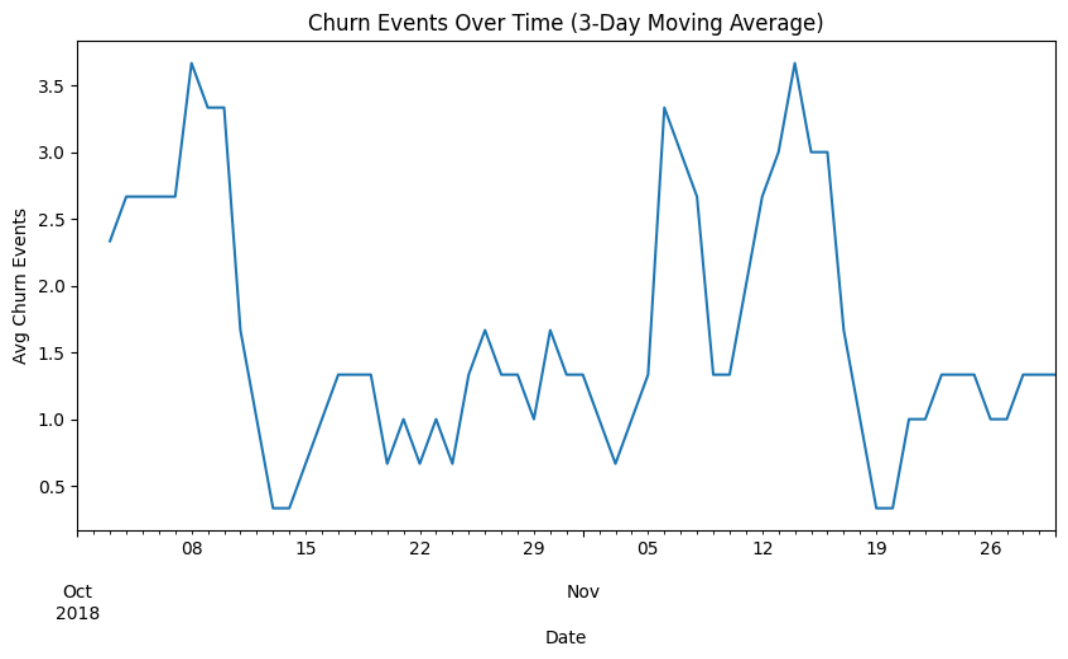
توزيع آخر نشاط لكل مستخدم بمرور الوقت

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



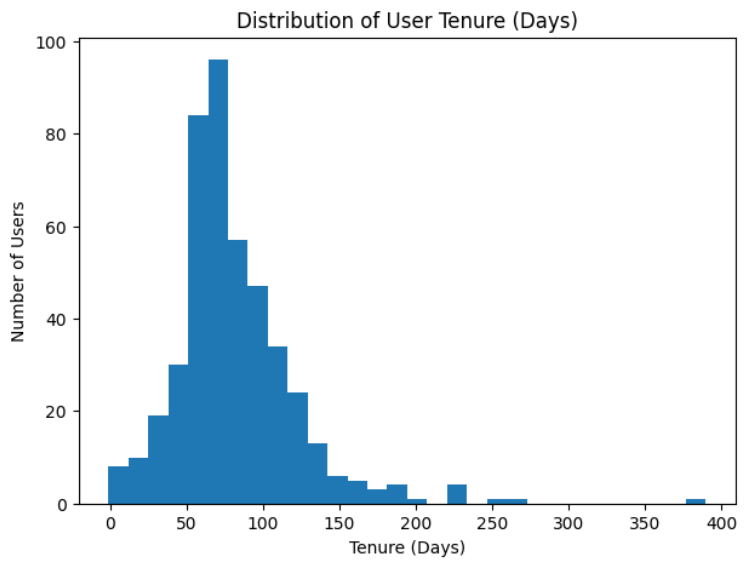
مقارنة عدد الأحداث بين المنسحبين وغير المنسحبين

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



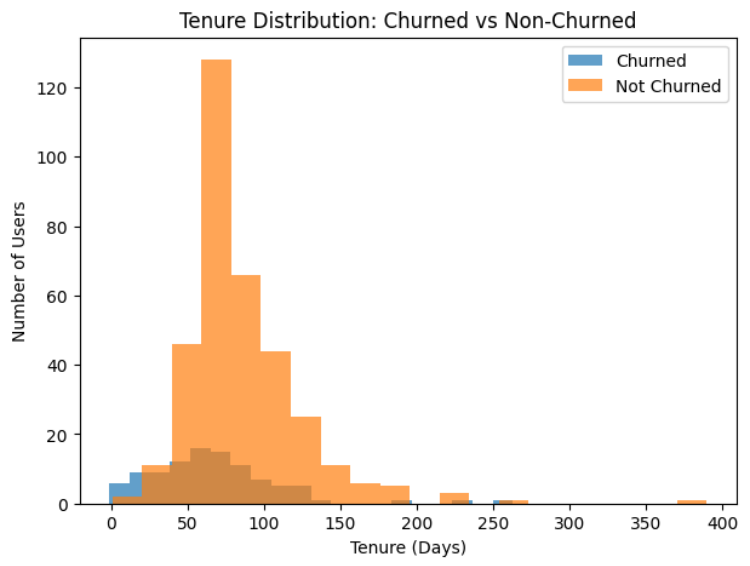
الانسحاب عبر الزمن

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



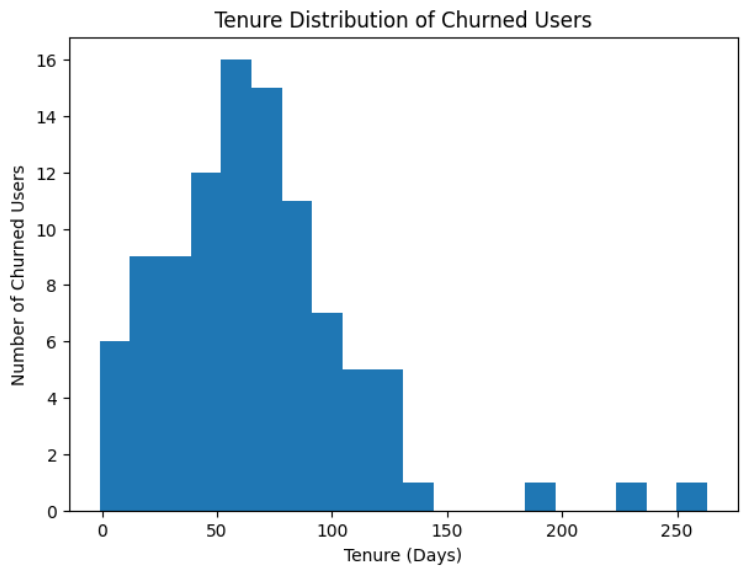
توزيع مدة بقاء المستخدم

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 المتغير من العوامل المهمة في التنبؤ بالإنسحاب .



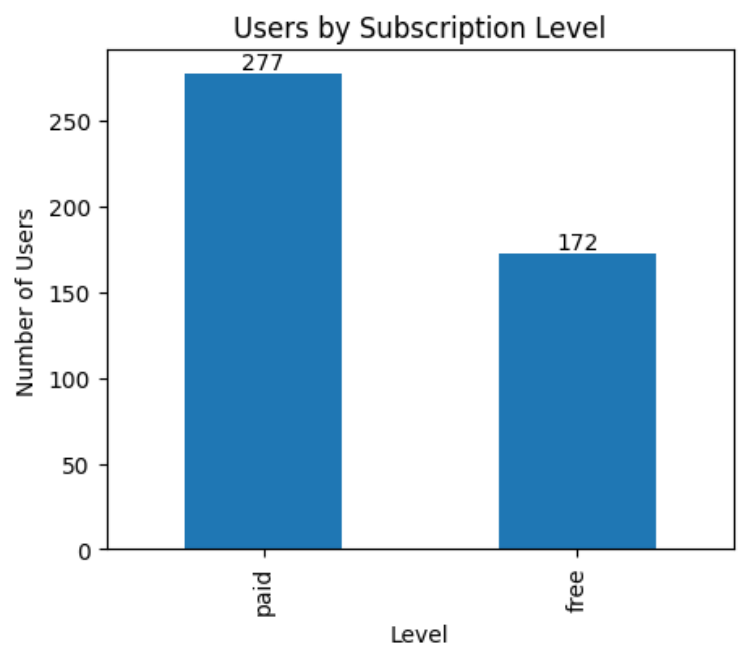
توزيع مدة بقاء المستخدم

المنسحبون لديهم مدة بقاء أقصر بوضوح في المقابل غير المنسحبين يميلون للبقاء لفترات أطول. نستنتج من هذا أن **Tenureمن أقوى المؤشرات على الانسحاب**.



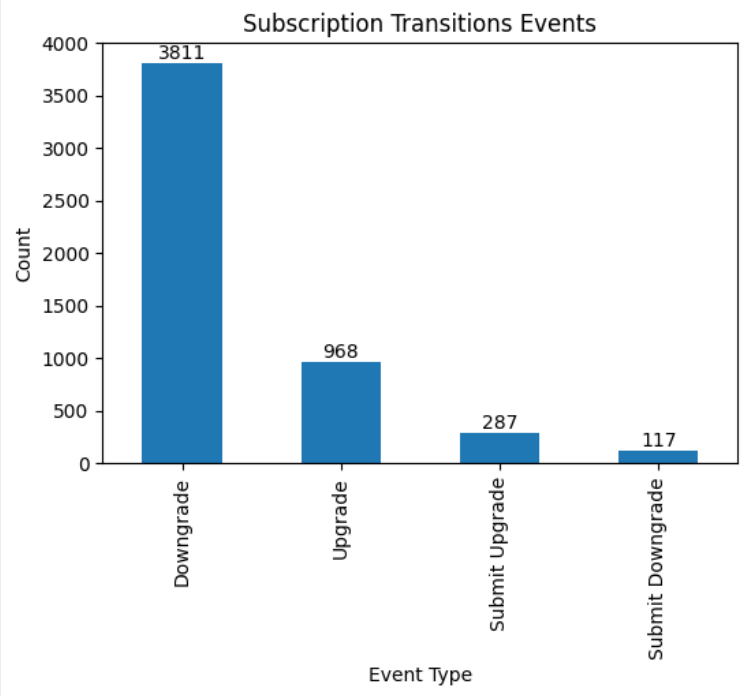
توزيع مدة بقاء المستخدمين المنسحبين

يمثل هذا الرسم توزيع مدة بقاء المستخدمين الذين قاموا بالانسحاب(Churned Users) ، حيث يوضح عدد الأيام التي ظل فيها المستخدم نشطًا في النظام قبل أن ينسحب .الغالبية العظمى من المستخدمين المنسحبين كانت مدة بقائهم أقل من **100يوم**. يوجد عدد محدود جدًا 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 لمراقبة سلوك الانسحاب لدى كلا الفئتين.



أحداث الانتقال بين مستويات الاشتراك

يظهر أن أحداث التخفيض (Downgrade) هي الأكثر تكرارًا مقارنة بالترقية، مما يعكس احتمالية وجود عدم رضا لدى المستخدمين ويجعل هذا العامل مؤشرًا مهمًا 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 المنسحبين وغير المنسحبين، مما سيتطلب:
  - اختيار مقاييس تقييم مناسبة بدل الاعتماد على الدقة فقط.
  - استخدام (Threshold Tuning)لتحسين أداء النموذج.
- المستخدم المنسحب يتميز بالخصائص التالية:
  - نشاط أقل داخل النظام.
  - عدد جلسات أقل مقارنة بالمستخدمين غير المنسحبين.
  - مدة بقاء أقصر في النظام.
  - تفاعل محدود مع الميزات الاجتماعية مثل Add Friend و.Thumbs Up
- قرار الانسحاب:
  - يحدث على مستوى المستخدم وليس على مستوى الحدث الفردي.
  - لا يعتمد على حدث واحد فقط مثلCancel Confirmation .
  - يتطلب تجميع سلوك المستخدم وتحليله عبر الزمن.
- نتائج التحليل الاستكشافي شكلت الأساس لـ:
  - تصميم الخصائص التنبؤية(Feature Engineering)
  - اختيار نموذج قابل للتفسير لتسهيل فهم أسباب الانسحاب.
  - ضبط (Threshold Tuning)لتحقيق توازن أفضل بين مؤشري Recall وPrecision

هندسة الخصائص

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

أهداف هندسة الخصائص

- تمثيل سلوك المستخدم على مستوى المستخدم بدلا 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** بدلا من الاعتماد على السجلات الفردية. Group by userId
- لكل مستخدم، تم استخراج ملخص شامل لسلوكه عبر فترة استخدامه للنظام.
- تم الاعتماد على نتائج EDA لاختيار الخصائص الأكثر ارتباطا 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 المتغير مستوى النشاط العام للمستخدم، حيث أن المستخدمين المنسحبين غالبًا ما يظهر لديهم عدد أقل من الأحداث.
num_sessions	عدد الجلسات الفريدة التي استخدم فيها المستخدم النظام. يدل هذا المتغير على مدى تكرار عودة المستخدم إلى الخدمة، ويُعد مؤشرًا مهمًا 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. يُعد مؤشرًا مباشرًا على استهلاك المحتوى الموسيقي واهتمام المستخدم بالخدمة الأساسية.
total_listen_time	إجمالي وقت الاستماع للمستخدم، ويعبّر عن الزمن الكلي الذي قضاه في الاستماع للمحتوى. هذا المتغير يعكس مستوى الارتباط بالخدمة بشكل أوضح من عدد الأغاني فقط.
Add Friend	عدد مرات استخدام المستخدم لميزة إضافة الأصدقاء. يعكس هذا المتغير مستوى التفاعل الاجتماعي داخل المنصة، وقد أظهر التحليل أن انخفاض هذا النوع من التفاعل يرتبط بارتفاع احتمالية الانسحاب.
Thumbs Up	عدد مرات الإعجاب بالمحتوى. يدل على رضا المستخدم عن المحتوى المقدم، ويُعد مؤشرًا إيجابيًا على تجربة المستخدم.
Thumbs Down	عدد مرات عدم الإعجاب بالمحتوى. 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 بالمحتوى والعودة إليه لاحقًا، وهو مؤشر على الاستمرارية.
last_level	آخر مستوى اشتراك للمستخدم (مجاني أو مدفوع) بناءً على آخر نشاط مسجل. يعبّر عن الحالة النهائية للمستخدم قبل الاستمرار أو الانسحاب.
num_downgrades	عدد مرات تخفيض الاشتراك من مدفوع إلى مجاني. يُعد 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 ارتباطا 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. يُستخدم لأغراض تحليلية وزمنية فقط، وليس كمدخل للنموذج .

بناء النماذج

تجهيز البيانات للنمذجة

كانت الخطوة الأولى في مرحلة النمذجة 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.

تقسيم البيانات(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 مجموعتي التدريب والاختبار
- يمنع تحيز النتائج ويعطي تقييمًا أكثر واقعية لأداء النموذج

نماذج التعلم الآلي المستخدمة

تمت تجربة عدة نماذج تصنيف بهدف اختيار النموذج الأنسب للتنبؤ بانسحاب المستخدمين . تم اختيار النماذج بناءً على شيوع استخدامها، قابليتها للتفسير، وقدرتها على التعامل مع بيانات غير متوازنة.

جدول النماذج المستخدمة وخصائصها

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

تقييم النماذج

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

مقاييس التقييم المستخدمة

- **Precision الدقة الإيجابية**

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

  يهتم بتقليل الإنذارات الخاطئة(False Positives)

- **Recallالاسترجاع – الأهم في هذا المشروع**

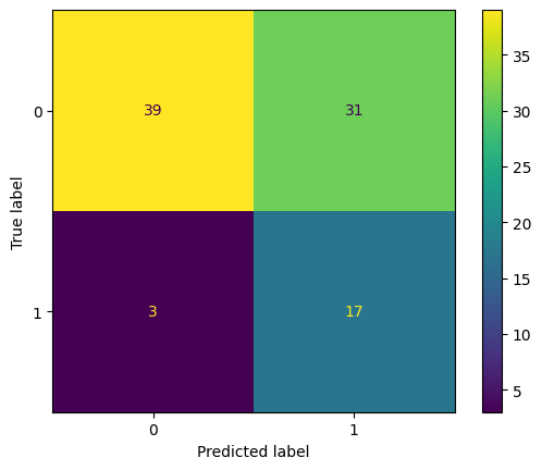
  يوضح قدرة النموذج على اكتشاف المستخدمين المنسحبين فعليًا، أي من بين جميع المستخدمين الذين انسحبوا بالفعل، **كم مستخدمًا نجح النموذج في التعرف عليه**

  يهتم بتقليل حالات الانسحاب غير المكتشفة(False Negatives)

- **F1-score**توازن بين Precision وRecall

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

تقييم نموذج Logistic Regression



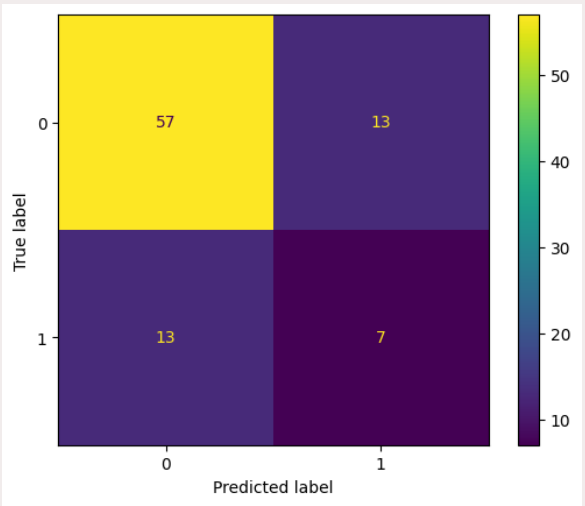
	precision	recall	f1-score	support
0	0.93	0.56	0.70	70
1	0.35	0.85	0.50	20
accuracy			0.62	90
macro avg	0.64	0.70	0.60	90
weighted avg	0.80	0.62	0.65	90

- النموذج نجح في اكتشاف معظم المستخدمين المنسحبين(Recall مرتفع)
- عدد حالات **False Negatives**منخفض جدا (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 سلبي حيث أنه من الممكن أن يكون لهولاء 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

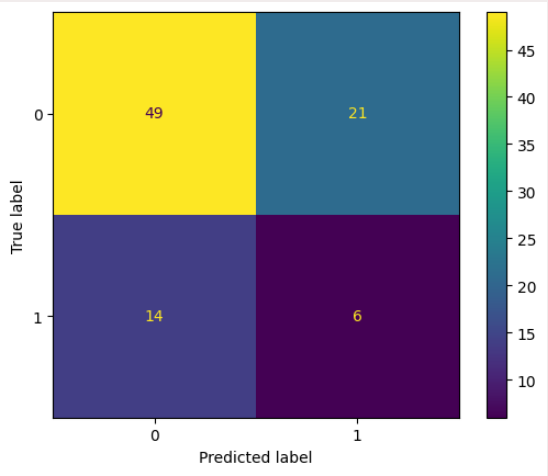
ROC-AUC: 0.5742857142857143

- 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

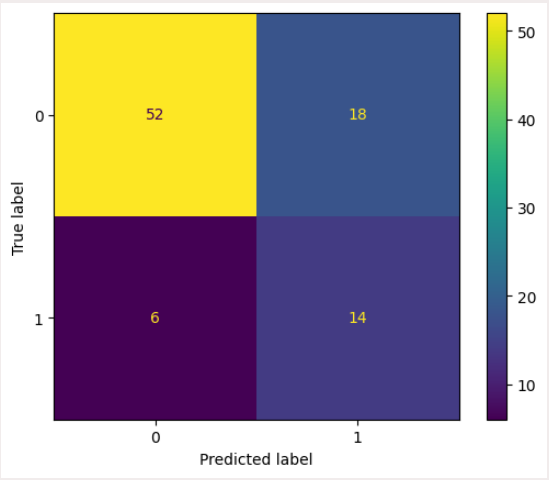
ROC-AUC: 0.5571428571428572

- أظهر أداءً محدودًا في اكتشاف المستخدمين المنسحبين.
- كل من Precision و Recall لفئة Churn منخفضان.
- عدد كبير من حالات الانسحاب لم يتم اكتشافها.

الاستنتاج:

تعقيد النموذج لم ينعكس على أداء أفضل في اكتشاف الانسحاب.

تقييم نموذج Logistic Regressionبعد 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

- انخفض عدد الإنذارات الخاطئة مقارنة بالنموذج الأساسي.
- لا يزال Recall مرتفعًا نسبيًا، مع تحسن واضح في Precision.

الاستنتاج:

ضبط العتبة أدى إلى نموذج أكثر توازنًا وقابلية للاستخدام العملي.

مقارنة نتائج النماذج

تم التركيز بشكل خاص على أداء الفئة المنسحبة (Churn = 1) نظرًا لأهمية اكتشاف حالات الانسحاب.

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

تم اختيار نموذج **Logistic Regression بعد ضبط العتبة** كنموذج نهائي، نظرًا لتحقيقه أفضل توازن بين مقاييس **Recall** و**Precision** مع الحفاظ على قدرة عالية على اكتشاف حالات الانسحاب. كما يتميز هذا النموذج ببساطته وقابليته للتفسير، مما يجعله الأنسب 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 المستخدم واستلام توقع الانسحاب بسهولة. ولضمان سهولة التشغيل والتوافق بين البيئات المختلفة، تم تغليف المشروع باستخدام **Docker** ليكون جاهزًا للتشغيل على أي جهاز دون الحاجة لإعدادات محلية معقدة.

```python
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)
```

```python
    # 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()}
```

```python
    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_serviceوالذي يحتوي على جميع الملفات اللازمة لتشغيل المشروع، بما في ذلك النموذج المدرَّب، ملفات الإعداد، وواجهة الـAPI.

تشغيل المشروع باستخدامDocker

1. فتح PowerShellأو Command Promptثم 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:

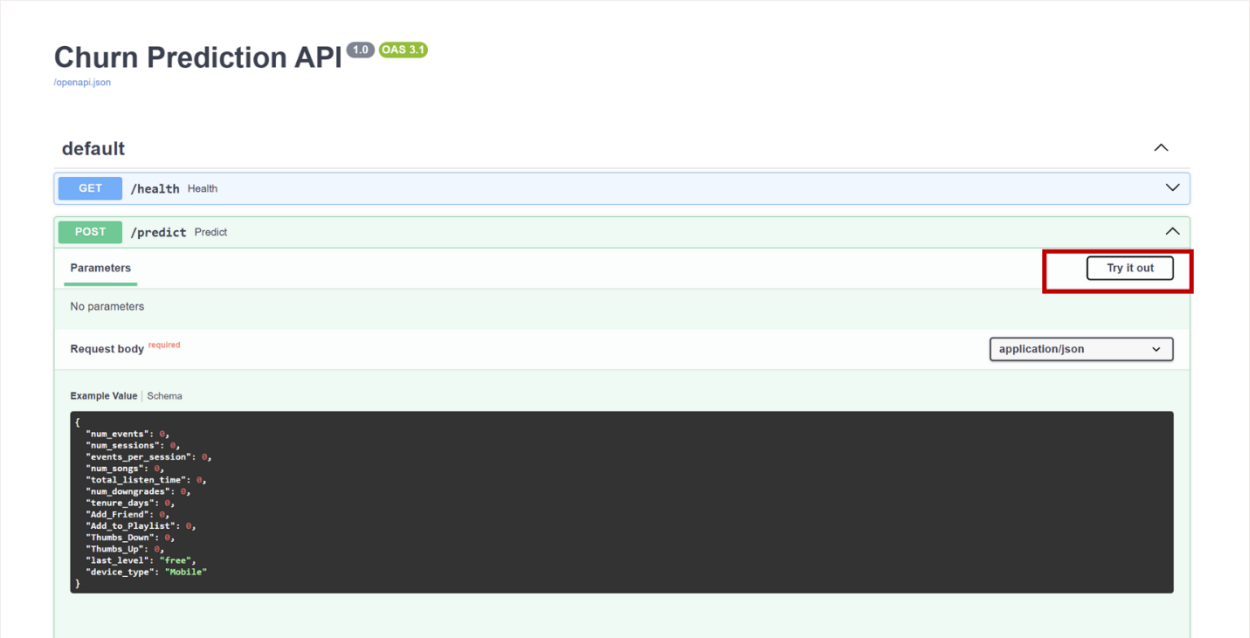
docker run -p 8000:8000 churn-api

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

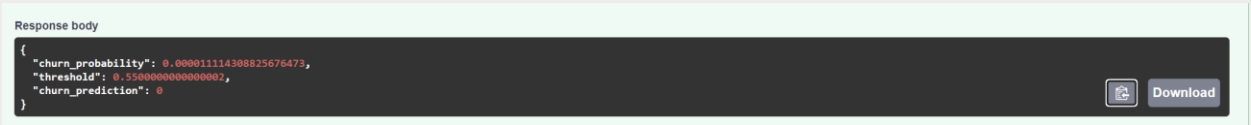
http://localhost:8000/docs

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

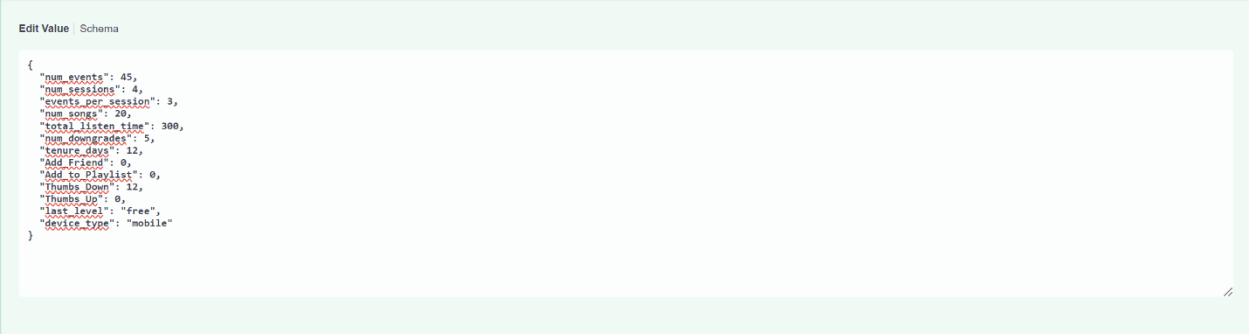
5. بعد الإنتقال للواجهة ستظهر هذه الصفحة ، لتجربة النموذج يجب الضغط على try it out

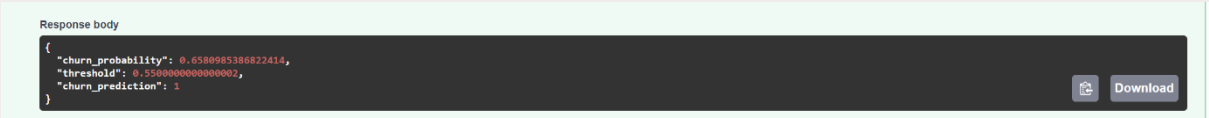


6. تعبئة البيانات ، في المثال أسفله قمت بإدخال بيانات تمثل مستخدمًا نشطًا، لديه تفاعل مرتفع ومدة بقاء أطول. نتيجة تنبؤ النموذج أظهرت أنه مستخدم غير منسحب واحتمالية انسحابه ضعيفة جدا.



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





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 (و أنا ايضًا أعمل بوظيفة ذات دوام كامل مما قلل الوقت أكثر) لم يكن بالإمكان إجراء تجارب أوسع على النماذج أو تنفيذ تحسينات إضافية كان من الممكن أن تؤدي إلى أداء أفضل للنموذج.

- **عدم توازن البيانات(Class Imbalance)**

عدد المستخدمين غير المنسحبين أكبر بكثير من المستخدمين المنسحبين، مما يؤثر على دقة التنبؤ بفئة الانسحاب ويتطلب تقنيات معالجة متقدمة.

- **حجم البيانات المستخدم محدود نسبيًا**

بعد ان تم تجميع البيانات على مستوى المستخدم اصبح حجمها محدود مما قيّد قدرة النموذج على التعلم من أنماط أكثر تنوعا للسلوك طويل 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 حقيقية**

إذ لم يتم تطبيق مراقبة مستمرة لأداء النموذج بعد النشر.(Model Monitoring)

التحسينات المستقبلي

يمكن تطوير هذا المشروع مستقبلًا وتحسين نتائجه من خلال:

- **تجربة نماذج أكثر تقدمًا**
  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 المعاملات(Hyperparameter Tuning)**
  لإيجاد الإعدادات المثلى لكل نموذج وتحقيق أداء أعلى.
- **إضافة نظام مراقبة للنموذج**
  لمتابعة:
  - تدهور الأداء
  - تغيّر سلوك المستخدمين(Data Drift & Concept Drift)
- **ربط النموذج بواجهة أمامية أو نظام حقيقي**
  لاستخدام التنبؤات في اتخاذ قرارات عملية مثل حملات الاحتفاظ بالمستخدمين.

الخاتمة

تم في هذا المشروع بناء نظام متكامل للتنبؤ بانسحاب المستخدمين اعتمادا على تحليل سلوكهم داخل المنصة. شمل العمل فهم البيانات وتحليلها استكشافيًا، 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 والاستدعاء.

أخيرًا، تم نشر النموذج باستخدام **FastAPI** وتغليفه عبر **Docker** ليكون جاهزًا للاستخدام والتشغيل في بيئات مختلفة، مما يجعل الحل قابلًا للتطبيق العملي والتوسع مستقبلًا.