

**ماجستير في علوم الحاسوب MCS**

**Loan bank approve\reject request**

**إشراف:**

**د.عصام سلمان**

|  |  |  |
| --- | --- | --- |
| الاسم | الرقم الجامعي | رقم الصف |
| سهى بسام سبسبي | **soha\_sabsaby\_289462** | **C2** |
| البتول محمد انور علي | **albatoul\_284575** | **C2** |
| مرام احمد شعبان | **Maram\_290313** | **C2** |
| يوسف ميشيل حداد | **youssef\_253209** | **C1** |
| دحام ياسر المحمد الخلف | **Dahaam\_285514** | **C2** |

**السؤال الرابع****:** حل مشكلات البيانات في هذا الجزء من المشروع، تم التركيز على معالجة مشكلات البيانات لضمان جودة النتائج وتحسين أداء النماذج المستخدمة لاحقًا. شملت عملية المعالجة عدة خطوات رئيسية، نوضحها كما يلي:

**1- تحميل البيانات واستكشافها:**

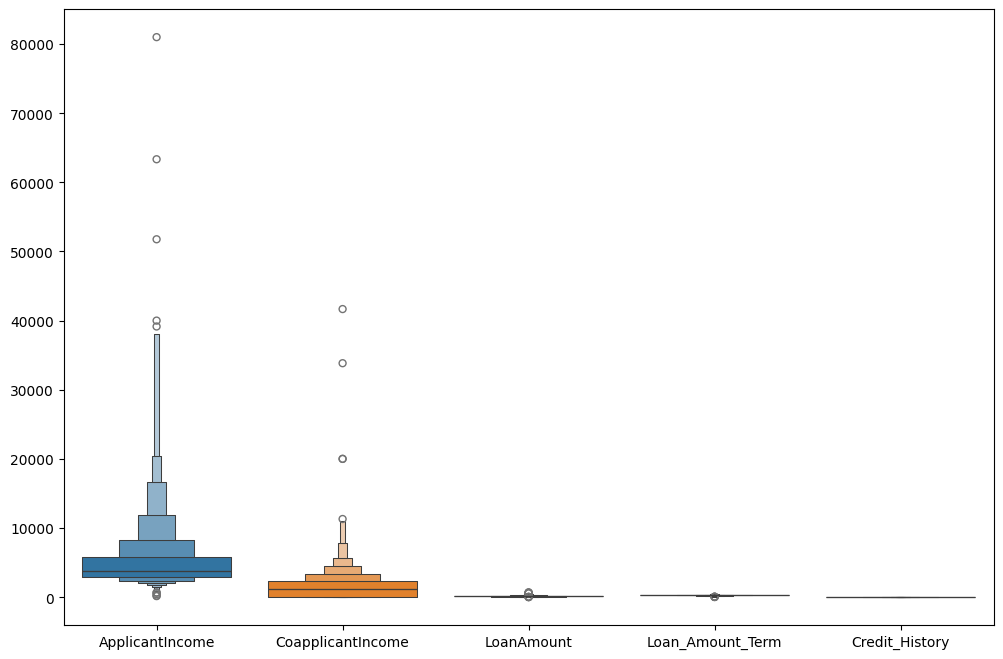
-بدأنا بتحميل مجموعة البيانات باستخدام مكتبة pandas- حيث تم عرض أول وآخر الصفوف لفهم البنية العامة للبيانات ومعرفة أنواع الأعمدة ومحتوياتها.

**2- فحص التكرارات:**

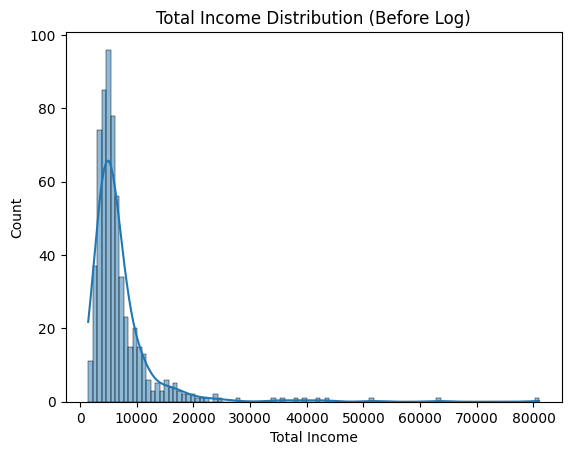
- تم التحقق من وجود صفوف مكررة باستخدام الدالة duplicated() . وقد لاحظنا عدم وجود اسطر مكررة.

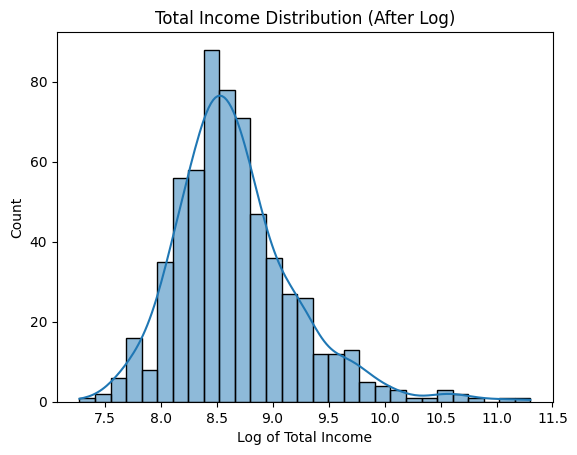
**3- اكتشاف القيم الشاذة :**

- تم استخدام الرسوم البيانية الصندوقية boxenplot لتحديد القيم الشاذة في الأعمدة الرقمية، ما يساعد على اتخاذ قرار بشأن التعامل معها أو تعديلها, حيث تم رصد قيم شاذة في دخل المتقدم ودخل شريك المتقدم وبقيمة القرض ومدة القرض ,حيث عملنا تطبيع وجمعنا دخل المتقدم وشريكه وأسميناه الدخل الكلي وطبقنا عليه التابع اللوغارتمي لكي يكون التوزيع أقرب للطبيعي ولتحقيق التباين, طبقنا أيضا التابع اللوغارتمي على بقية القيم الشاذة مثل قيمة القرض ومدة مبلغ القرض و وضحنا بالرسم الفرق بتوزيع البيانات قبل تطبيق التحويل اللوغرتمي وبعده وبينا الانحراف.



سوف نعرض لقطات شاشة توضح الميلان في مخطط الدخل الكلي و الفرق بعد معالجتها:





**4- التعامل مع القيم المفقودة:**

- حيث انه تم التعامل مع القيم المفقودة بشكل خاص وذلك لاختلاف أنواع البيانات حيث أنه يوجد قيم فئوية وعادة تتم معالجتها باستخدام القيم الاكثر تكرارا وقيم رقمية حيث تعالج عادة بالمتوسط والوسيط سوف يتم استعراضها جميعا مع كيفية المعالجة للملأ.

**LoanAmount** تم استخدام الوسيط (median) لملء القيم المفقودة لتقليل تأثير القيم المتطرفة.

**Loan\_Amount\_Term**تم استخدام المتوسط (mean) لملء القيم المفقودة.

**Credit\_History** في البداية تم محاولة استخدام المتوسط، لكن الناتج كان قيمة غير صالحة (0.84) نظرًا لأن العمود يحتوي فقط على قيم ثنائية (0 و1)، لذلك تم استبدال القيم المفقودة بالقيمة الأكثر تكرارًا (mode) للحصول على نتيجة منطقية وواقعية.

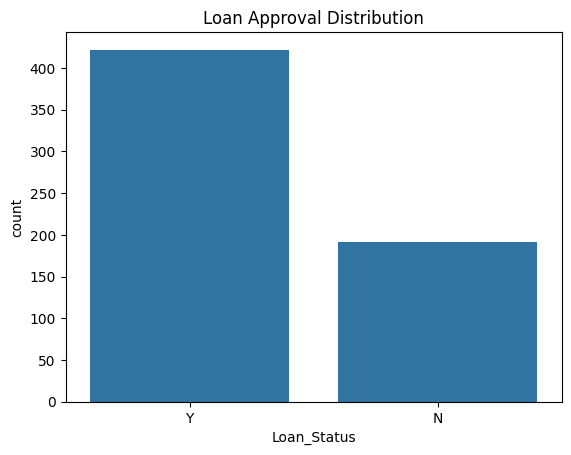
-**الأعمدة الفئوية (Categorical Columns)** مثل: Gender, Married, Dependents, Self\_Employed، تم ملء القيم المفقودة فيها باستخدام القيمة الأكثر تكرارًا (mode).

**5-عرض التوزيعات العامة:**

- تم استخدام الرسوم البيانية لعرض توزيع الفئات في الأعمدة الفئوية مثل (Gender, Married ,Education, Loan\_Status,

- تم توليد الرسوم باستخدام seaborn countplot() لكل عمود , حيث سوف نعرض عمود حالة القرض

مخطط الأعمدة (countplot) لعمود Loan\_Status يوضح حالة وجود عدم توازن كبير (Imbalanced Data):



#### ****6 -تحليل التوزيع (Histograms + KDE):****

- تم رسم مدرجات تكرارية histplot مع منحنيات الكثافة kde=True للخصائص الرقمية مثل:

* + ApplicantIncome
  + CoapplicantIncome
  + LoanAmount

#### ****7 -تحليل الارتباط (Correlation):****

- تم توليد خريطة حرارية (Heatmap) للارتباط بين المتغيرات الرقمية.

- أوضحت الخريطة وجود ارتباط متوسط وقوي بين:

* + ApplicantIncom و LoanAmount ≈ 0.57

**8- ترميز البيانات الفئوية :**

-تم تحويل القيم النصية في الأعمدة الفئوية إلى قيم رقمية باستخدام LabelEncoder، لتجهيزها للاستخدام في النماذج لاحقًا، حيث أن معظم خوارزميات التعلم الآلي لا تقبل البيانات النصية

**9- حذف الأعمدة الغير ضرورية:** حذفنا بعض الأعمدة مثلا التي قمنا بتطبيق تطبيع عليها وبعض الأعمدة الأخرى غير الضرورية وذلك لتقليل عدد المتغيرات وتبسيط النموذج.

بعض الملاحظات مما سبق:

* من خلال تحليل العمود Loan\_Status و Credit\_History، لوحظ عدم توازن كبير في عدد الحالات المقبولة مقابل المرفوضة.
* هذا الخلل تمت معالجته لاحقًا باستخدام StratifiedKFold أثناء بناء النماذج لضمان تمثيل عادل لكافة الفئات في كل تقسيم.
* نلاحظ أن المتغير Credit\_History له تأثير كبير على قرار الموافقة على القرض. وبالرغم من أن الدخل يلعب دورًا مهمًا، إلا أن سجل الائتمان الثابت والمستقر يعتبر عاملاً أكثر أهمية من دخل مرتفع غير مضمون , وهذا يتماشى مع منطق العمل المصرفي، حيث يُفضل العملاء ذوي الدخل المنتظم والسجل الائتماني الجيد لضمان القدرة على السداد.

**بناء وتدريب النموذج :**

الأدوات المستخدمة : منصة الحوسبة ,Jupyter notebook ومكتبة بايثون Scikit-learn.

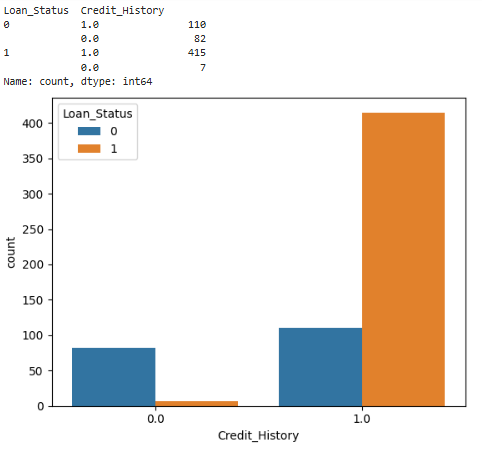
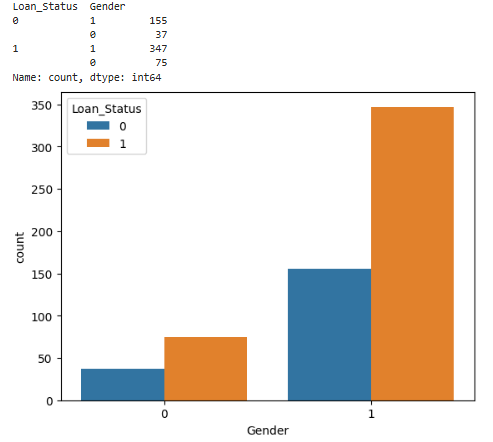
يتضمن هذا القسم أربعة مراحل :

1. تقسيم البيانات.
2. اختبار النماذج.
3. اختيار النموذج الأدق.
4. تصدير النموذج.
5. تقسيم البيانات :

بعد أن تم تصحيح البيانات سابقاً لتكون جاهزة لتجربتها بالنماذج، تم اختيار الطريقة Cross\_Validation عوض عن Train\_Test\_Split , لما تقدم الأولى من تقديرأكثر دقة لقدرة النموذج على التعميم, حيث تقسم البيانات الى أقسام متعددة ويدرب النموذج بشكل متكرر على مجموعات متعددة من هذه الأقسام، وأيضاً مناسب لحجم البيانات غير الكبير لدينا.

وضمن الطريقة Cross\_Validation تم اختيار النوع Stratified KFold لتوريع البيانات.

كما تبين سابقا وجود توزيع غير متوازن للبيانات :



ي

تم اختيار Stratified KFold لأنه يضمن أن يحافظ كل Fold على نفس نسب الفئات في مجموعة البيانات الأصلية, وهذا مفيد لمجموعات البيانات غير المتوازنة لدينا.

تم إعطاء المتغير X قيم بيانات الأعمدة ((Features Columns, والمتغير y تم إعطائه قيم عمود حالة القرض Loan\_Status.

تم استدعاء من المكتبة sklearn.model\_selection الطريقة StratifiedKFold والمقيم

cross\_val\_score .

معاملات StratifiedKFold :

تم اختيار عدد التقسيمات n\_splits = 10 لأن حجم البيانات لدينا صغيرفنحتاج عدد تقسيم أكبر ليكون النموذج معمم أكثر لحالة البيانات.

بما أن البيانات لدينا مبعثرة ولاتحتاج اختيارات عشوائية, فأبقينا المعاملات random\_state و shuffle بقيمها الافتراضية.

1. اختبار النماذج :

تم اختيار النماذج الثلاث التالية :

1. Logistic Regression : نموذج يوفر طريقة قوية لتحليل النتائج الثنائية ( كما بحالة بياناتنا) والتنبؤ بها من خلال نمذجة احتمالات الفئات المختلفة استنادًا إلى العلاقات بين ميزات الإدخال X والمتغير المستهدف y .
2. Decision Tree Classifier : نموذج يقوم بالتنبؤ خطوة بخطوة حيث يقسم البيانات لمجموعات فرعية. يمكنه التقاط علاقات غير خطية معقدة بين السمات والمتغير المستهدف.
3. Random Forest Classifier : نموذج يعمل على تدريب أشجار قرار متعددة على مجموعات فرعية عشوائية مختلفة من بيانات التدريب وخصائصه, يُساعد هذا على تقليل ال Overfitting وتحسين الدقة مقارنةً باستخدام شجرة قرار واحدة.

الكود المستخدم لتقسيم البيانات, تدريب أحد النماذج, طباعة قيم الدقة والمقاييس للنموذج :



1. اختيار النموذج الأدق :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Average**  **F1 score** | **Average**  **recall** | **Average**  **precision** | **Average**  **Accuracy** | **Model** |
| **0.88** | **0.98** | **0.79** | **0.81** | **Logistic Regression** |
| **0.80** | **0.80** | **0.79** | **0.72** | **Decision Tree** **Classifier** |
| **0.86** | **0.94** | **0.80** | **0.79** | **Random Forest Classifier** |

تم اختيار النموذج Logistic Regression بناء على قيم الدقة ومقاييس الأداء المبينة بالجدول,

حيث أعطى الدقة الأكبر من بين النماذج المدروسة.

1. تصدير النموذج :

تم استخدام وحدة pickle المدمجة في بايثون لحفظ نموذج التعلم المختار, لإعادة تحميله إلى الذاكرة لإجراء تنبؤات أو تحليلات إضافية، دون الحاجة إلى إعادة تدريب النموذج.

