

Machine Learning

أستاذ ومنسق المقرر : د. عصام سلمان

إعداد الطلاب:

abdulrahman_290065

rand_293968

julia_298202

في إطار سعي المؤسسات المالية والبنوك لتحسين وتبسيط عملية اتخاذ القرارات المتعلقة بالموافقة أو الرفض لطلبات القروض، يهدف هذا المشروع إلى تطوير منصة ويب تدعم اتخاذ هذه القرارات بشكل سريع وفعال من خلال تحليل البيانات واستخدام تقنيات تعلم الآلة. سيستعمل الموقع على مجموعة من الميزات الأساسية مثل إضافة وإزالة الطلبات، بالإضافة إلى عرض تحليلات استكشافية للبيانات وحل المشكلات المتعلقة بجودة البيانات.

تتمثل النقطة الرئيسية في هذا المشروع في تطوير نموذج ذكاء اصطناعي باستخدام تقنيات تعلم الآلة بهدف تدريب نموذج قوي وفعال على مجموعة بيانات محددة. يتضمن هذا المشروع تصميم وتنفيذ عملية تدريب منظم تبدأ من استيراد البيانات ومعالجتها، ثم بناء النموذج وتدريبه، وأخيراً تصميم موقع ويب يتضمن API للتنبؤ بحالة القروض، إدارة الطلبات، وعرض تقارير تحليلية. إذاً الخطوة الأولى هي تحديد الخوارزمية الأفضل لبناء النموذج باستخدام مكتبات Python المتخصصة في تعلم الآلة.

(1) التحضير المبدئي للبيانات :

- i. استيراد المكتبات الأساسية لتحليل البيانات
- ii. قراءة البيانات
- iii. عرض أول خمس صفوف

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

data = pd.read_csv("../content/LoanApprovalPrediction.csv")

[ ] data.head(5)
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	LP001002	Male	No	0.0	Graduate	No	5849	0.0	NaN	360.0	1.0	Urban	Y
1	LP001003	Male	Yes	1.0	Graduate	No	4583	1508.0	128.0	360.0	1.0	Rural	N
2	LP001005	Male	Yes	0.0	Graduate	Yes	3000	0.0	66.0	360.0	1.0	Urban	Y
3	LP001006	Male	Yes	0.0	Not Graduate	No	2583	2358.0	120.0	360.0	1.0	Urban	Y
4	LP001008	Male	No	0.0	Graduate	No	6000	0.0	141.0	360.0	1.0	Urban	Y

iv. تحويل المتغيرات الفئوية إلى عددية : في هذه المرحلة، قمنا بمعالجة المتغيرات الفئوية الموجودة في مجموعة البيانات لتحويلها إلى قيم عددية قابلة للاستخدام من قبل نماذج تعلم الآلة.

- a. تحديد الأعمدة الفئوية.
 - b. تطبيق LabelEncoder على كل عمود فئوي.
 - c. إعادة التحقق من عدد الأعمدة الفئوية المتبقية.
- بعد التحويل، يتم التحقق من عدم وجود أعمدة فئوية متبقية تهيئة البيانات لتصبح جاهزة للنماذج الإحصائية والخوارزميات التي لا تدعم القيم النصية مباشرة.

```
[ ] # Import label encoder
from sklearn import preprocessing

# label_encoder object knows how
# to understand word labels.
label_encoder = preprocessing.LabelEncoder()
obj = (data.dtypes == 'object')
for col in list(obj[obj].index):
    data[col] = label_encoder.fit_transform(data[col])

[ ] # To find the number of columns with
# datatype==object
obj = (data.dtypes == 'object')
print("Categorical variables:",len(list(obj[obj].index)))
```

(2) تقسيم البيانات إلى تدريب واختبار:

بعد تجهيز البيانات وتحويل القيم الفئوية إلى قيم عددية، قمنا بما يلي :

- i. فصل الميزات عن المتغير المستهدف .
- ii. تقسيم البيانات إلى مجموعة تدريب واختبار.
- iii. التحقق من الأشكال النهائية للمجموعات.

الهدف هو تمكين النماذج من التعلم على جزء من البيانات (التدريب)، ثم اختبار مدى قدرتها على التعميم باستخدام جزء لم تره من قبل (الاختبار).

```
from sklearn.model_selection import train_test_split

X = data.drop(['Loan_Status'],axis=1)
Y = data['Loan_Status']
X.shape,Y.shape

X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
                                                    test_size=0.4,
                                                    random_state=1)
X_train.shape, X_test.shape, Y_train.shape, Y_test.shape
```

((358, 11), (240, 11), (358,), (240,))

(3) تقييم أولى للنماذج على بيانات التدريب:

بعد تجهيز البيانات وتجزئتها، قمنا بتطبيق أربعة نماذج تصنيف مختلفة لتقييم أدائها على مجموعة التدريب فقط. الهدف من هذه الخطوة هو الحصول على نظرة أولية حول قدرة النماذج على التعلم من البيانات، وليس بالضرورة الحكم على فعاليتها النهائية، النماذج المستخدمة :

a. **Random Forest Classifier** : تم استخدام 7 أشجار مع معيار التقسيم entropy.

b. **K-Nearest Neighbors (KNN)** : بعدد جيران = 3 .

c. **Support Vector Classifier (SVC)** : تم استخدام الإعدادات الافتراضية .

d. **Logistic Regression (LR)** : أيضاً تم استخدام الإعدادات الافتراضية .

تم طباعة نسبة الدقة لكل نموذج على بيانات التدريب، وكانت النتائج كما يلي:

```
[ ] from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression

from sklearn import metrics

knn = KNeighborsClassifier(n_neighbors=3)
rfc = RandomForestClassifier(n_estimators = 7,
                             criterion = 'entropy',
                             random_state = 7)

svc = SVC()
lc = LogisticRegression()

# making predictions on the training set
for clf in (rfc, knn, svc,lc):
    clf.fit(X_train, Y_train)
    Y_pred = clf.predict(X_train)
    print("Accuracy score of ",
          clf.__class__.__name__,
          "=",100*metrics.accuracy_score(Y_train,
                                          Y_pred))
```

Accuracy score of RandomForestClassifier = 98.04469273743017
Accuracy score of KNeighborsClassifier = 78.49162011173185
Accuracy score of SVC = 68.71508379888269
Accuracy score of LogisticRegression = 79.60893854748603

(4) التقييم الفعلي للنماذج على مجموعة الاختبار:

في هذه المرحلة، قمنا بتقييم أداء النماذج الأربعة التي تم تدريبها مسبقًا على مجموعة الاختبار هذا التقييم يمثل المؤشر الحقيقي لمدى قدرة كل نموذج على التعميم والتعامل مع بيانات جديدة غير مرئية.

```
[ ] # making predictions on the testing set
for clf in (rfc, knn, svc, lc):
    clf.fit(X_train, Y_train)
    Y_pred = clf.predict(X_test)
    print("Accuracy score of ",
          clf.__class__.__name__, "=",
          100*metrics.accuracy_score(Y_test,
                                     Y_pred))
```

```
Accuracy score of RandomForestClassifier = 82.5
Accuracy score of KNeighborsClassifier = 63.74999999999999
Accuracy score of SVC = 69.16666666666667
Accuracy score of LogisticRegression = 80.83333333333333
```

(5) تحليل النتائج:

من خلال نتائج التجربة، تبين أن خوارزمية **Random Forest Classifier** قدمت أداءً متميزًا مقارنة ببقية الخوارزميات المستخدمة في تصنيف البيانات. يتجسد ذلك في قدرتها على تحقيق توازن جيد بين **التعلم والتعميم**، مما يجعلها من الخيارات المثلى في المشكلات التي تتطلب تصنيفًا دقيقًا على مجموعات بيانات متنوعة.

- **دقة النموذج على مجموعة التدريب** : حقق النموذج دقة تصل إلى 99%، مما يشير إلى قدرة الخوارزمية على التعلم بشكل فعال من البيانات المدخلة.
- **دقة النموذج على مجموعة الاختبار** : حيث أن دقة النموذج على مجموعة الاختبار كانت 82.5%، تعتبر هذه النسبة مرتفعة، وتعكس قدرة الخوارزمية على تعميم المعرفة المستخلصة من بيانات التدريب.
- بناءً على هذا التحليل، سيتم اختيار **خوارزمية Random Forest** كالنموذج الأمثل للتنبؤ بإمكانية قبول القرض، بناءً على مجموعة من المتغيرات (الخصائص) المتعلقة بالمتقدمين. ستشمل هذه العملية التعامل مع البيانات المفقودة، تطبيق تقنيات معالجة القيم الشاذة، ثم ترميز البيانات الفئوية، وذلك لتدريب النموذج على تصنيف المتقدمين إلى فئتين رئيسيتين:
 - **قرض معتمد (Loan Approved)** .
 - **قرض مرفوض (Loan Rejected)** .

❖ تحليل البيانات وتدريب نموذج التنبؤ بالقروض باستخدام خوارزمية Random Forest (train_model.py) :

(1) استيراد المكتبات:

- **pandas** : لإدارة البيانات وتحليلها.
- **numpy** : لتنفيذ العمليات الرياضية المتقدمة.
- **os** : لإدارة المسارات والملفات.
- **Joblib** : لحفظ النماذج المدربة لتستخدمها لاحقًا.
- **sklearn.model_selection** : لاستخدام دالة `train_test_split` لتقسيم البيانات.
- **sklearn.preprocessing** : لاستخدام `LabelEncoder` لتحويل المتغيرات الفئوية إلى أرقام.
- **sklearn.ensemble** : لاستخدام نموذج `RandomForestClassifier` (غابة عشوائية) للتصنيف.

(2) إعداد مسارات البيانات والنماذج:

- يتم تحديد مسارات الملفات التي تحتوي على البيانات والنموذج المُدرَّب باستخدام `os.path.join` .

(3) تحميل وتنظيف البيانات :

- **تحميل البيانات** : يتم تحميل ملف `loan_prediction.csv` الذي يحتوي على بيانات القروض باستخدام `pandas.read_csv()` .
- **التعامل مع القيم المفقودة** : يتم ملء القيم المفقودة في الأعمدة المختلفة باستخدام:
 - القيم الأكثر تكرارًا في الأعمدة الفئوية (`mode()`) .
 - القيمة الافتراضية (مثل 360 في `Loan_Amount_Term`) ، أو القيمة المتوسطة (مثل `LoanAmount`)

- إزالة القيم الشاذة باستخدام **IQR** : يتم استخدام مقياس المدى بين الرباعيات (Interquartile Range - IQR) لاكتشاف القيم الشاذة في بعض الأعمدة مثل : LoanAmount, ApplicantIncome, CoapplicantIncome واستبدال القيم الشاذة بالقيم المتوسطة لتلك الأعمدة.

(4) التحضير والترميز :

- إزالة الأعمدة غير الضرورية : مثل عمود Loan_ID لأنه غير مفيد في عملية التدريب.
- معالجة الأعمدة الفئوية:
 - يتم استبدال القيم غير الرقمية في العمود Dependents (مثل "+3") بالقيمة الرقمية 3.
 - يتم ترميز الأعمدة الفئوية (Gender, Married, Education, Self_Employed) باستخدام LabelEncoder لتحويلها إلى قيم عددية لتناسب النموذج.

(5) تقسيم البيانات وتدريب النموذج :

- تقسيم البيانات : يتم تقسيم البيانات إلى مدخلات (x) وأهداف (y) ، حيث أن x يحتوي على الخصائص مثل الدخل والحالة الاجتماعية، بينما يحتوي y على القيمة المستهدفة وهي حالة القرض (Loan_Status) .
- تقسيم البيانات إلى تدريب واختبار : باستخدام train_test_split ، يتم تقسيم البيانات إلى مجموعة تدريب (80%) واختبار (20%) .
- معالجة عدم التوازن : يتم حساب الأوزان العكسية لكل فئة بناءً على تواترها في مجموعة التدريب لتقليل تأثير البيانات غير المتوازنة.
- تدريب النموذج : يتم تدريب نموذج الغابة العشوائية باستخدام البيانات المدربة (X_train, Y_train) .
- حفظ النموذج والمُشفّرات : بعد تدريب النموذج، يتم حفظه مع المُشفّرات باستخدام مكتبة **joblib** لنتمكن من استخدام النموذج لاحقاً. يتم حفظ النموذج في ملف (model.pkl)، والمُشفّرات في ملف (encoders.pkl) .

(6) تنفيذ العملية:

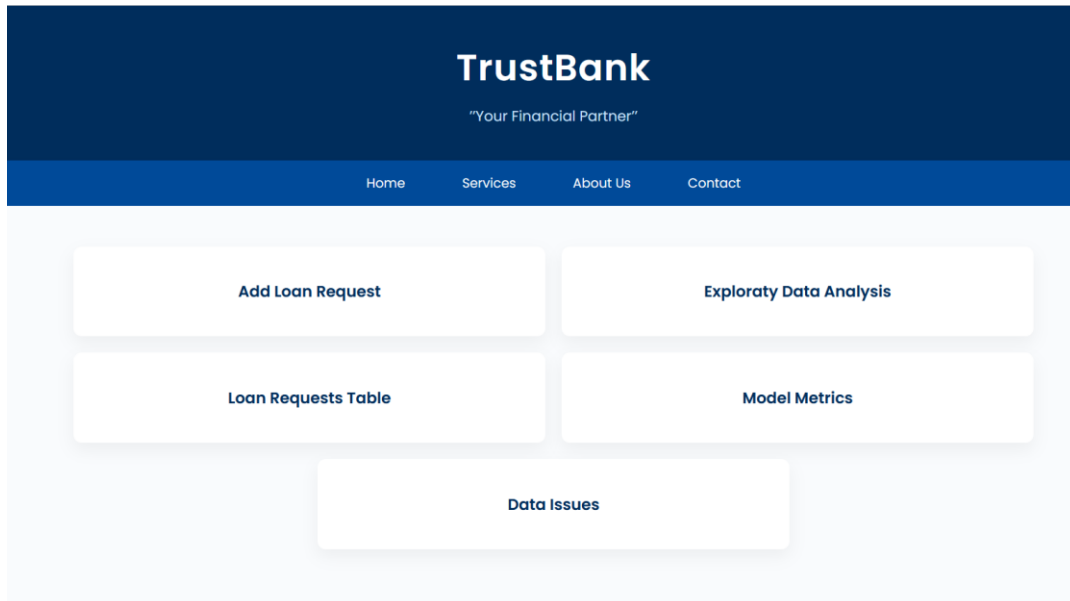
- عند تنفيذ الكود، يتم استدعاء دالة train_and_save_model() ، والتي تقوم بجميع الخطوات السابقة من تحميل البيانات، تنظيفها، تدريب النموذج، وحفظه.

(7) النتيجة النهائية:

- يتم طباعة رسالة تأكيد بعد حفظ النموذج والمُشفّرات بنجاح :النموذج والمُشفّرات تم حفظها بنجاح ✓ .
- وبذلك قد أصبح النموذج جاهزاً للاستخدام في تنبؤ إمكانية قبول أو رفض القروض بناءً على خصائص المتقدمين، مما يساهم في تحسين عملية اتخاذ القرار داخل المؤسسة المالية.

❖ تصميم وبناء الموقع الإلكتروني Trust Bank :

➤ الواجهة الأمامية Frontend : (التقنيات المستخدمة : HTML ,CSS ,JavaScript)



• الصفحة الرئيسية (Home Page) :

الصفحة الرئيسية لموقع TrustBank هي الواجهة الأولى ، تحتوي على المكونات الأساسية التي تسهل التنقل داخل الموقع ، وهي عبارة عن خمسة بطاقات رئيسية ، كل بطاقة تحتوي على رابط إلى صفحة مختلفة في الموقع:

- إضافة طلب قرض Add Loan Request
- جدول طلبات القروض Loan Requests Table
- تحليل البيانات الاستكشافي Exploraty Data Analysis
- معالجة البيانات Data Issues Report
- مقاييس النموذج Model Metrics

• صفحة طلب القرض (Add Loan Request) :

تُعد صفحة "طلب القرض" في موقع Trust Bank واحدة من الصفحات الحيوية التي تتيح تقديم طلبات القروض ، تتضمن النموذج الذي يتطلب تعبئة مجموعة من البيانات لتحديد أهلية المتقدم للحصول على القرض . من خلال الضغط على زر "Predict" ، يتم عرض النتيجة التي تحدد ما إذا كان القرض مقبولاً أم مرفوضاً استناداً إلى نموذج تعلم الآلة المستخدم كما يتم عرض احتمالية الموافقة كنسبة مئوية ، وبالضغط على زر الإرسال "Submit" يتم إرسال البيانات لتخزين طلب القرض في قاعدة البيانات ، وفي حال وجود خطأ في البيانات المدخلة ، سيتم عرض رسالة تحذير تطلب من المستخدم تعبئة كافة الحقول .

• صفحة جدول طلبات القروض (Loan Requests Table) :

تُعتبر جزءاً أساسياً لعرض جميع طلبات القروض التي تم تقديمها عبر الموقع. توفر الصفحة واجهة تفاعلية لعرض تفاصيل كل طلب، وتتيح للمسؤولين إمكانية حذف أي طلب قرض من النظام إذا لزم الأمر.

• صفحة التحليل الاستكشافي (Exploraty Data Analysis) :

تعتمد الصفحة على جلب ملخص البيانات من الواجهة البرمجية الخارجية، وتعرض إحصائيات مثل عدد الطلبات ومتوسط القروض والدخل ونسبة الموافقة. كما تتضمن رسوماً بيانية توضيحية باستخدام Chart.js، مدعومة بمكتبة chartjs-plugin-datalabels لإظهار النسب بدقة داخل المخططات.

• صفحة تقرير معالجة البيانات (Data Issues Report) :

تعرض هذه الصفحة تقريراً عن مشاكل جودة البيانات مثل القيم المفقودة والقيم الشاذة، يتم جلبها من الواجهة البرمجية الخارجية API ، تُعرض المشاكل في جداول، ويتبعها قسم يوضح طرق المعالجة المستخدمة مثل التعويض أو الحذف في النهاية ، كما تُعرض ملاحظات ختامية حول معالجة البيانات وجودتها.

• صفحة مقاييس أداء نموذج تعلم الآلة المستخدم (Model Metrics) :

تعرض صفحة Model Metrics أداء نموذج تعلم الآلة من خلال أربعة مؤشرات رئيسية: الدقة (Accuracy) ، الدقة الإيجابية (Precision) ، الاسترجاع (Recall) ، ودرجة F1 (F1 Score) . يتم جلب هذه القيم تلقائياً من الواجهة البرمجية الخارجية عند فتح الصفحة، وتُعرض بشكل مبسط وواضح لتقييم جودة النموذج.

➤ الواجهة الخلفية Backend والربط مع نموذج التنبؤ :

تم بناء الواجهة البرمجية باستخدام FastAPI لإدارة طلبات القروض وربطها بنموذج التنبؤ ، فيما يلي شرح مختصر مع ذكر المكتبات والدوال الأساسية لكل مرحلة :

1. تصميم البيانات و API :

- المكتبة: Pydantic
- الدالة الأساسية BaseModel : لتصميم نموذج البيانات.
- تم تعريف نموذج بيانات موحد للتحقق من صحة البيانات المدخلة وتسهيل التواصل عبر نقاط نهاية API (Endpoints) باستخدام FastAPI .

2. تحميل النموذج والمشفرات:

- المكتبة: joblib
- الدالة الأساسية joblib.load() : لتحميل النموذج المدرب والمشفرات مثل LabelEncoder.
- يتم تحميل النموذج المدرب والمشفرات لتحويل القيم النصية إلى تمثيلات رقمية.

3. معالجة البيانات:

- المكتبة: scikit-learn , Pandas
- الدالة الأساسية fillna() : لتعويض القيم المفقودة، LabelEncoder() لتحويل القيم النصية، و drop() لحذف الأعمدة غير المفيدة.
- تتم معالجة البيانات لتعويض القيم المفقودة وتحويل النصوص إلى أرقام، وحذف الأعمدة غير المهمة.

4. التحليل الاستكشافي للبيانات (EDA) :

- المكتبة: Matplotlib Seaborn
- الدالة الأساسية: plt.plot(), sns.histplot() : لإنشاء الرسوم البيانية التحليلية.
- يتم عرض تحليلات مرئية لتوزيع البيانات مثل التنبؤات، الجنس، الدخل، والمنطقة السكنية.

5. التنبؤ بحالة القرض:

- المكتبة: **scikit-learn**
- الدالة الأساسية: `model.predict()`: لإجراء التنبؤ باستخدام النموذج المدرب.
- يتم تمرير البيانات للنموذج المدرب لتحديد ما إذا كان الطلب سيُقبل أو يُرفض مع تحديد نسبة الاحتمالية.

6. تقييم النموذج:

- المكتبة: **scikit-learn**
- الدالة الأساسية: `classification_report()` , `accuracy_score()` لتقييم دقة النموذج باستخدام مقاييس مثل الدقة والاسترجاع و **F1**.
- يتم تحليل نتائج التنبؤات لتقييم جودة أداء النموذج.

7. تكامل النظام:

- المكتبة: **FastAPI**
 - الدالة الأساسية: `@app.get()` , `@app.post()` لتعريف نقاط النهاية الخاصة بال-API.
 - يتم تنظيم النظام في وحدات مستقلة تُستدعى عبر API مما يسهل الصيانة والتطوير.
 - يمكن الاطلاع على الواجهة البرمجية للموقع على الرابط التالي: <https://mlt.vilmo.app/docs#/>
- وبذلك أصبح الموقع يدير ويحلل طلبات القروض باستخدام **FastAPI** , **scikit-learn** , **Pandas**، ويقدم التنبؤات مع تحليلات مرئية باستخدام **Matplotlib** , **Seaborn**، مع ضمان دقة الأداء وجودة النتائج.

➤ الاختبار والنتائج :

- إضافة طلب قرض وإظهار نتيجة التنبؤ : حيث قمنا بإدخال بيانات افتراضية ثم الضغط على زر Predict ، وكانت النتيجة على الشكل التالي :

Loan Request Form

Please fill in the form below to request a loan:

Prediction Result: **Rejected**

Approval Probability: 46%

Gender	Married
<div>Male</div>	<div>No</div>
Dependents	Education
<div>0</div>	<div>Graduate</div>
Self Employed	Applicant Income
<div>Yes</div>	<div>120</div>
Coapplicant Income	Loan Amount
<div>2500</div>	<div>120</div>
Loan Term (months)	Credit History
<div>240</div>	<div>Good (1)</div>
Property Area	
<div>Rural</div>	

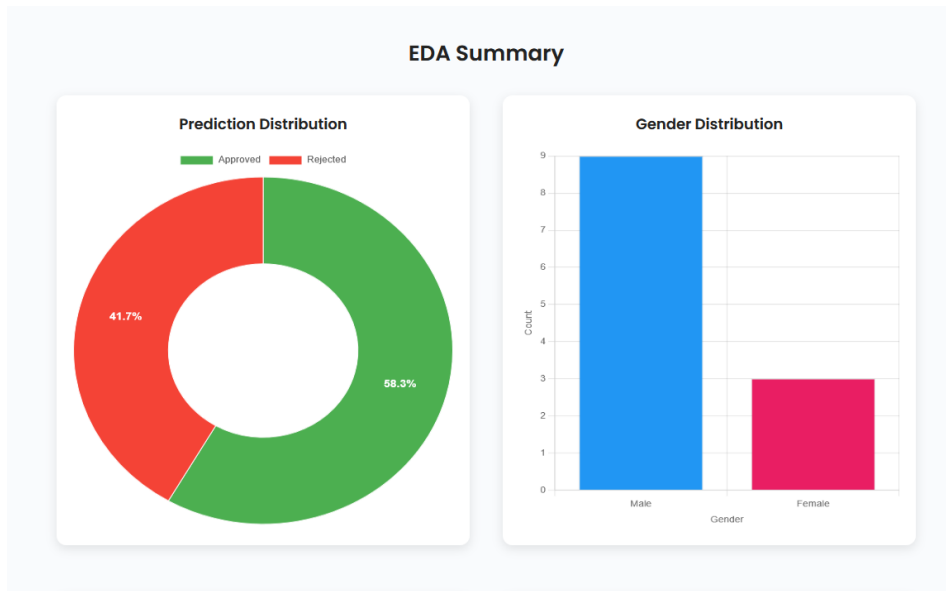
Predict

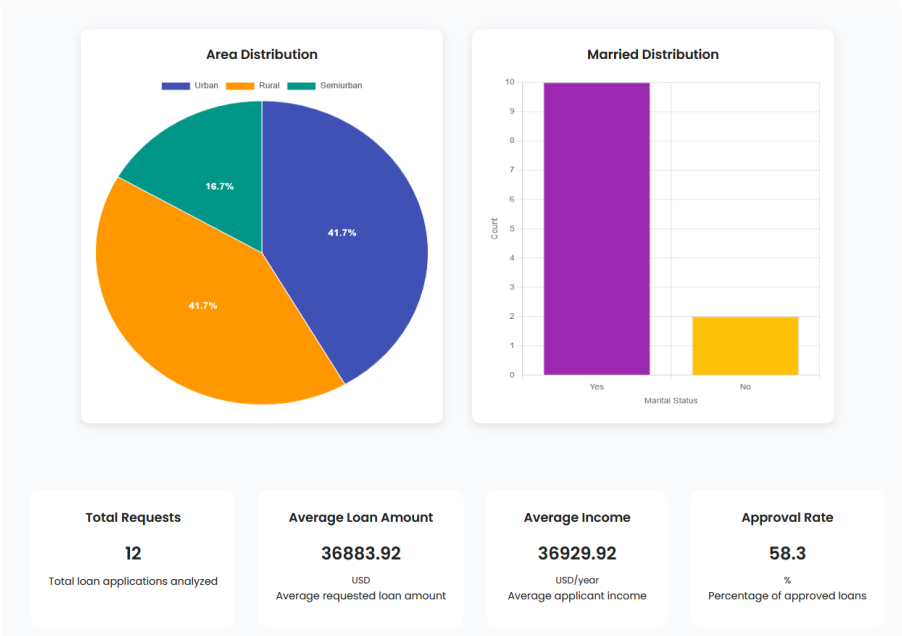
Submit

- عرض جدول الطلبات مع إمكانية حذف الطلب : حيث قمنا بإضافة 12 طلب إلى قاعدة البيانات ، وذلك بإدخال البيانات والضغط على زر Submit .

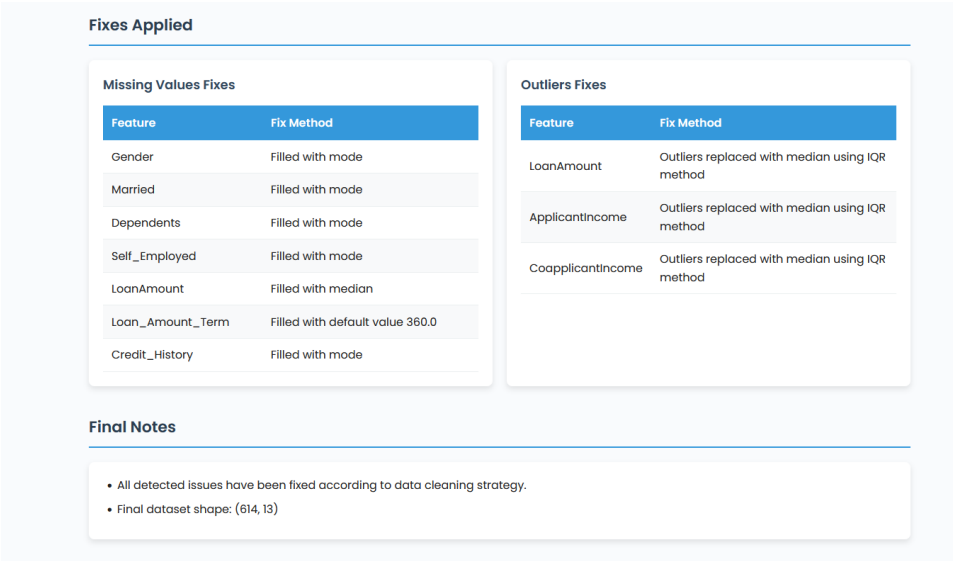
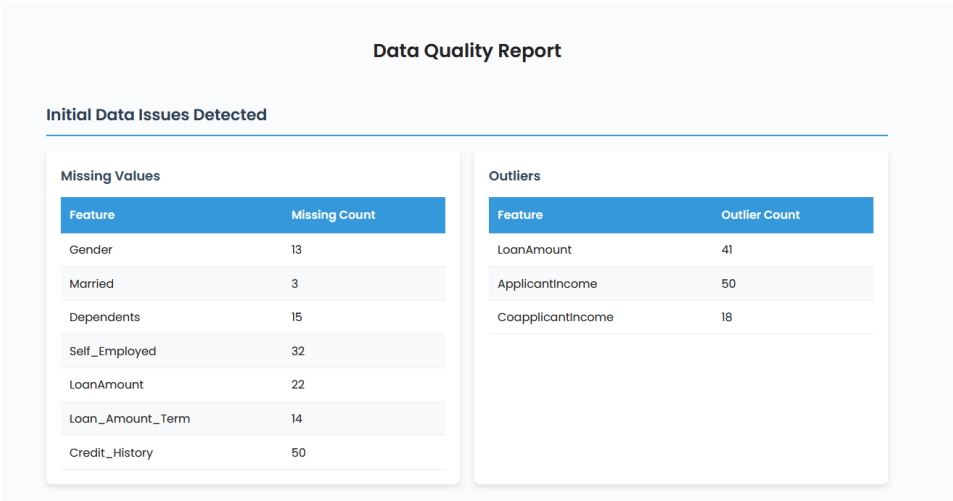
TrustBank													
"Your Financial Partner"													
Home Services About Us Contact													
Loan Requests													
Loan_ID	Gender	Married	Dependents	Education	Self_Employed	Applicant_Income	Coapplicant_Income	Loan_Amount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status	Actions
b004435d-3918-4438-8b7c-77325ea9209f	Male	Yes	1	Graduate	No	5000	0	150	360	1	Urban	Approved	Delete
1b448761-0d18-49f7-9e35-6d7a72f0a024	Male	Yes	1	Graduate	No	5000	0	150	360	1	Urban	Approved	Delete
ff065db6-a0e9-4392-bced-edee6d897edd	Male	Yes	0	Graduate	Yes	0	4859	127	360	1	Rural	Approved	Delete
1b17009f-ac21-45f0-8b76-1b8a7d138b73	Male	Yes	0	Graduate	Yes	0	127	127	360	1	Rural	Rejected	Delete
49fb6948-3cd6-4f14-a340-8e607f8600bf	Male	Yes	0	Graduate	Yes	0	127	127	360	1	Rural	Rejected	Delete
1525f8f1-a5a6-4197-b718-bae976b54cc2	Male	Yes	0	Graduate	Yes	0	127	127	360	1	Rural	Rejected	Delete
034f272f-7306-4d60-a7af-95eb607d5f8a	Male	Yes	0	Graduate	Yes	20000	20000	20000	360	1	Urban	Approved	Delete
a0a5e2bf-b70d-4f23-b56c-dfe0067ac879	Male	Yes	3+	Graduate	No	400000	400000	400000	12	1	Urban	Approved	Delete
9d67534e-2820-4f87-8b53-7bbfd78a974d	Male	Yes	0	Graduate	Yes	4259	0	1247	360	1	Rural	Approved	Delete

- تحليل البيانات الاستكشافي للطلبات المدخلة : حيث تعرض هذه الصفحة بعض التحليلات الاستنتاجية للطلبات المدخلة إلى قاعدة البيانات ، مثل : معدل القبول والرفض ، عدد الذكور والإناث المتقدمين ، نسبة توزع مناطق المتقدمين جغرافياً ، الحالة الاجتماعية ، كما تعرض عدد الطلبات ومعدل دخل المتقدم وحجم القرض . و بناءً على الطلبات التي قمنا بإدخالها ظهرت النتائج التالية :





- **صفحة التقرير الخاصة بمعالجة البيانات الشاذة والمفقودة :** تعرض هذه الصفحة تقريراً يوضح المشكلات التي واجهتنا أثناء تحضير البيانات (قيم مفقودة وقيم شاذة) عند تدريب النموذج ، كما توضح كيفية معالجة هذه المشكلات .



- **صفحة عرض مقاييس أداء النموذج :** تعرض هذه الصفحة مقاييس أداء النموذج المُدرَّب المستخدم في عملية التنبؤ بقبول أو رفض طلب المقترض .
حيث كانت **الدقة Accuracy** (النسبة المئوية للعينات التي صنفت بشكل صحيح) مساوية **76%** ، كما أعطى نسبة **دقة إيجابية Precision** (من بين كل ما تم تصنيفه على انه "إيجابي" كم كان صحيحاً فعلاً ؟) مساوية **75%**، أما بالنسبة **للاسترجاع Recall** (من بين كل الإجابات الحقيقية في البيانات ، كم منها تم اكتشافه ؟) فأعطى نسبة **94%** .
مقياس F1 (المتوسط التوافقي بين Precision و Recall) يساوي **83%** .

Model Performance Metrics

Accuracy:	0.76
Precision:	0.75
Recall:	0.94
F1 Score:	0.83

➤ الخاتمة :

في الختام يمكننا التأكد على أن تقنيات تعلم الآلة أداة أساسية للمؤسسات المالية لتحسين دقة قراراتها وتقليل المخاطر. لأنها تساهم في تعزيز الكفاءة التشغيلية وتوفير الوقت، مما يعزز قدرة المؤسسة على التكيف والنمو في بيئة اقتصادية ديناميكية.