تقرير مشروع الكشف عن الاحتيال المالي

أولاً: التقرير الفني للمشروع

:نظرة عامة على النظام

يهدف هذا المشروع إلى تطوير نظام ذكي للكشف عن الاحتيال المالي باستخدام مجموعة بيانات PaySim1 "الاصطناعية. تم تصميم النظام كـ "خط أنابيب متكامل لتعلم الآلة، مصمم لتصنيف (Pipeline) المعاملات المالية بدقة عالية بين معاملات شرعية أو احتيالية.

:المنهجية المتبعة

اعتمد المشروع على دورة حياة كاملة لتعلم الآلة (ML Lifecycle)، تشمل المراحل التالية:

- معالجة البيانات: استخراج البيانات الخام، تنظيفها، وإزالة المعرفات غير الضرورية، مع ابتكار ميزات جديدة تعتمد على فرق الرصيد (Balance-difference features).

التدريب والتحسين: تدريب نموذج -
مع استخدام تقنية RandomForestClassifier
للوصول إلى أفضل المعاملات Optuna
لرفع كفاءة النموذج (Hyperparameters).

التكامل والتوثيق: استخدام - MLflow لتتبع
التجارب وتسجيل النماذج، و DVC لضمان قابلية
استنساخ العمليات.

النشر: تقديم التوقعات عبر واجهة برمجة تطبيقات -
(FastAPI) وواجهة مستخدم تفاعلية
(Streamlit).

نتائج الأداء:

أظهر النموذج أداءً استثنائياً في الكشف عن الاحتيال،
مما يجعله مثالياً للاستخدام في أنظمة الفرز الفوري.

- الدقة (Accuracy): 0.9967

- درجة F1 (F1 Score): 0.9598

- الدقة النوعية (Precision): 0.9263

- الاستدعاء (Recall): 0.9957

ثانياً: التوثيق التقني للأدوات والتقنيات

يعتمد هذا النظام على بيئة عمل متكاملة باستخدام لغة Python كركيزة أساسية لبناء خط الأنابيب، الواجهة البرمجية، ولوحة التحكم.

1. معالجة البيانات والنمذجة:

- **Pandas & NumPy:** للتحميل، التنظيف، إجراء العمليات الحسابية الرقمية (كاللوغاريتمات)، وهندسة الميزات.

- **Scikit-learn:** لإجراء عمليات التقسيم والتدريب، بالإضافة إلى استخدام RandomForestClassifier كنموذج تصنيف نهائي.

- **LabelEncoder:** لتحويل القيم النصية في أنواع المعاملات إلى قيم رقمية قابلة للمعالجة.

2. التحسين وإدارة التجارب:

- **Optuna:** للبحث التلقائي عن أفضل إعدادات النموذج لرفع الأداء.

- **MLflow:** لإدارة دورة حياة النموذج، تتبع التجارب، وتخزين الإصدارات لضمان القابلية للتكرار.

- **DVC:** لإدارة خطوط أنابيب البيانات بشكل قابل للاستنساخ.

3. النشر والواجهات:

- **FastAPI & Uvicorn:** لنشر النموذج كخدمة سريعة، مع استخدام Pydantic للتحقق من صحة البيانات الواردة (REST API).

- **Streamlit & Requests:** لتوفير لوحة تحكم سهلة الاستخدام للمستخدم النهائي، حيث تقوم بربط واجهة المستخدم بخدمة التوقع Requests.

- **Matplotlib:** لتصور وتحليل النتائج عبر مصفوفات الارتباك ومنحنيات التعلم.