

MASTER OF SCIENCE IN ARTIFICIAL INTELLIGENCE — MIDOCEAN UNIVERSITY, UAE

AI-BASED SMART ACCOUNTING SYSTEM



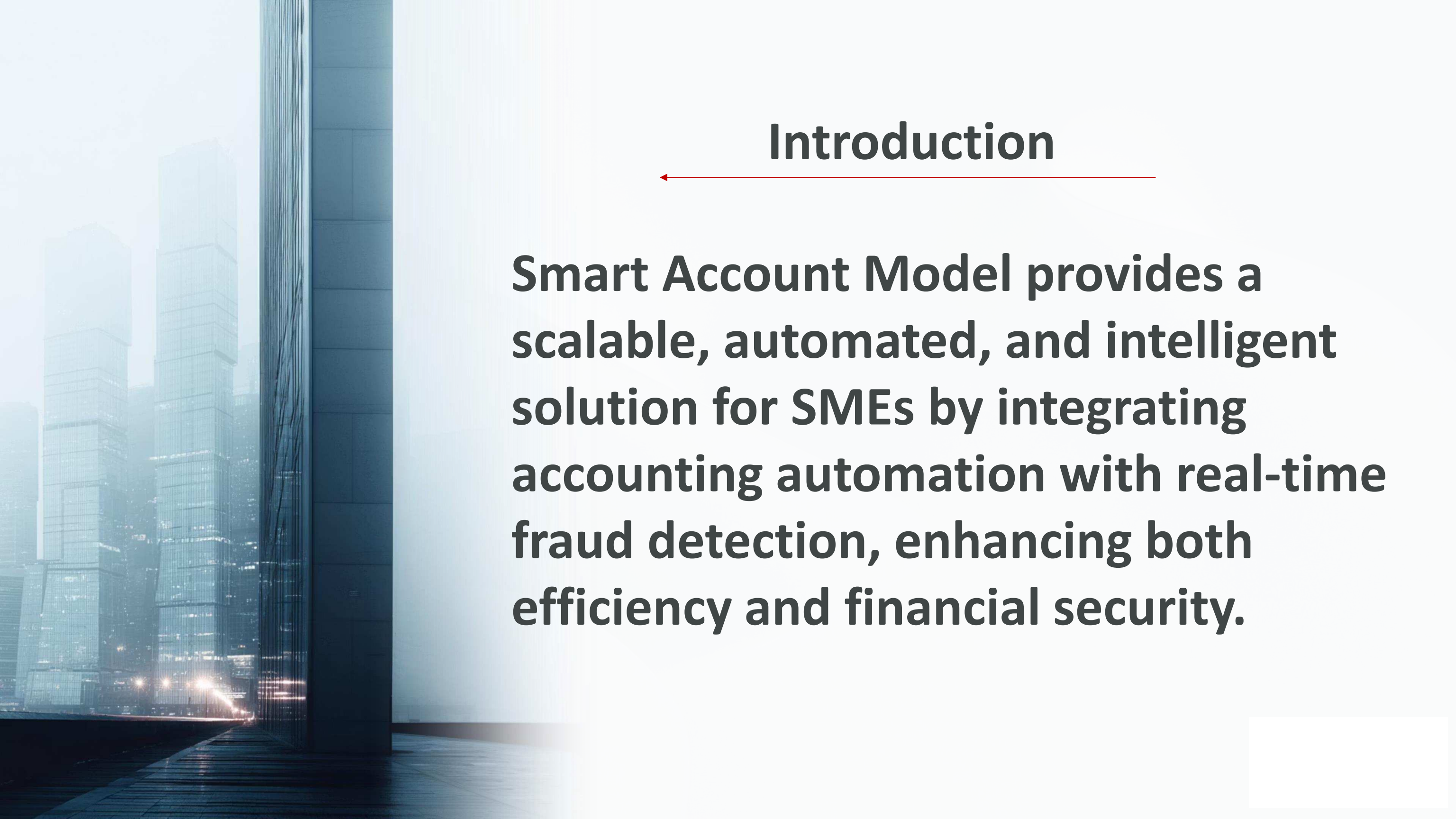
to Automate Accounting and Fraud Detection in SMEs

Prepared by: Jihad Abdullh Najee Alhuraiby ID No.: 253002789

Supervisor: Dr. Amjad Qtaish — College of Informatics,
Midocean University, Comoros



Introduction & Research Framework



Introduction

Smart Account Model provides a scalable, automated, and intelligent solution for SMEs by integrating accounting automation with real-time fraud detection, enhancing both efficiency and financial security.

Problem Definition



- **Manual accounting is error-prone**
- **SMEs vulnerable to fraud**
- **Traditional methods fail in real-time detection**



Objectives

- Automate accounting tasks
- Detect fraud using ML/DL
- Evaluate using datasets
- Provide explain ability (XAI)





Background

- **AI Transforms Accounting**
- **Automation + Fraud Detection**
- **Improves Accuracy & Efficiency**



Fraud in SMEs

- **5% Revenue loss Annually**
- **Weak Controls**
- **Manual Systems Ineffective**



Challenges

- **Data imbalance**
- **Limited SME datasets**
- **Privacy issues**
- **Explain ability needed**

RELATED STUDIES & LITERATURE REVIEW (2021–2026)

Ref (author) year	Paper Title	Description	Dataset	Methods	Results	Advantage	Disadvantage	Almalki, F. et al. (2025)	Financial Fraud Detection Using Explainable AI and Stacking Ensemble Methods	Stacking ensemble with SHAP & LIME for fraud detection	IEEE-CIS Fraud Detection (590k transactions)	Stacking (XGBoost, LightGBM, CatBoost) + XAI	99% Accuracy, 0.99 AUC	High performance + excellent explainability	Focused mainly on credit card data
Yang, H. et al. (2026)	Study of Artificial Intelligence for Financial Fraud Detection: Datasets, Methods, and Challenges	Comprehensive review of AI methods, datasets, and challenges in financial fraud detection	Public fraud datasets	Systematic literature review	Comparative metrics (Accuracy, AUC, F1)	Broad coverage of current trends	Limited focus on SME accounting data	Siam, A.M. et al. (2025)	Hybrid feature selection framework for enhanced credit card fraud detection	Hybrid filter + embedded feature selection framework	Credit card datasets	Hybrid FS + ML classifiers	Improved accuracy & reduced features	Efficient feature reduction	Not tested on SME accounting data
Abdo-Salloum, A.M. et al. (2026)	The Role of Artificial Intelligence in Transforming Accounting Practices	Systematic review on AI in accounting automation and fraud prevention	Literature review	Thematic analysis	Qualitative benefits	Strong emphasis on SME benefits	No empirical model testing	Cai, M. (2025)	Research on the automation of intelligent accounting information processing based on deep learning	NeuroLedger-Net for automated accounting and anomaly detection	Kaggle financial datasets	LSTM + Autoencoders + MLP	96.3% Accuracy	Strong in accounting automation	Limited use of XAI
Attaufiq, A.Y. et al. (2026)	Design and Development of an AI-Based Smart Accounting System	Case study on AI smart accounting for internal control and fraud prevention	Company case data (Indonesia)	ML models + case study	Confusion Matrix	Practical real-world SME implementation	Limited generalizability (single case)	Alruwaili, T.F. et al. (2025)	The impact of artificial intelligence on accounting practices	Impact of AI on efficiency and fraud detection in SMEs	Literature review	Systematic review	Qualitative improvements	Good SME relevance	Purely theoretical
Chen, Y. (2025)	Deep Learning in Financial Fraud Detection: Innovations, Challenges, and Applications	Systematic review of 108 studies on DL in fraud detection	Various public datasets	Systematic review + DL models	AUC, Accuracy, F1-score	Deep technical insights on DL	No new proposed model	Jin, J. et al. (2025)	The analysis of fraud detection in financial market under stacking ensemble	Stacking ensemble with dynamic weighting	Real financial market data	Stacking (LR, DT, RF, SVM, NN)	High accuracy, low false alarm rate	Innovative dynamic weighting	Lacks explainable AI

RELATED STUDIES & LITERATURE REVIEW (2021–2026)

Kibriya, M.G. et al. (2025)	Explainable AI (XAI) Transparency for Financial Fraud Detection	Application of SHAP and LIME for model transparency	Public fraud datasets	ML models + SHAP + LIME	Good interpretability metrics	High transparency	No automation module
Moura, L. et al. (2025)	AI and Financial Fraud Prevention : Mapping the Trends and Institutional Implications	Bibliometric mapping of AI trends in fraud prevention	Literature	Bibliometric + literature review	Trend analysis	Identifies institutional gaps	No technical implementation
Mienye, I.D. et al. (2023)	A Machine Learning Method with Hybrid Feature Selection for Credit Card Fraud Detection	Hybrid filter-wrapper feature selection	Credit card datasets	Information Gain + Genetic Algorithm	Improved model performance	Effective hybrid FS technique	Not adapted for SME tabular data
Zhang, Z. et al. (2022)	Financial Fraud Identification Based on Stacking Ensemble Learning Algorithm	Stacking using textual + financial data	Financial reports + transaction data	Stacking ensemble	Superior to baselines	Integration of unstructured data	Not real-time SME focused
Hamal, S., & Senvar, O. (2021)	Comparing the performance of ML classifiers in detecting financial accounting	Performance comparison of 7 ML classifiers on Turkish SMEs	341 Turkish SMEs financial data	ML classifiers (RF best)	Best results with ensembles	Highly relevant to SMEs	Old (2021), no DL or XAI

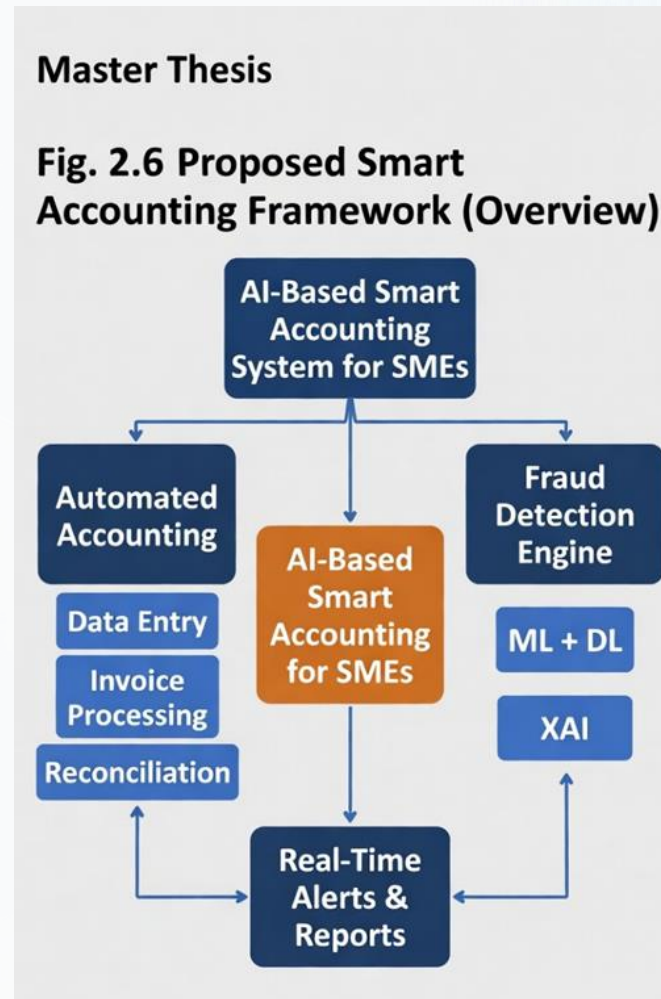
	fraud for SMEs						
Zhu, X. et al. (2021)	Intelligent financial fraud detection practices in post-pandemic era	Systematic review of post-pandemic fraud detection practices	Various	Systematic literature review	Qualitative + model comparison	Highlights real-time needs	No new proposed system



Architecture & Methodology

Smart Accountent Model

- AI-based smart accounting system
- Automation + fraud detection + XAI



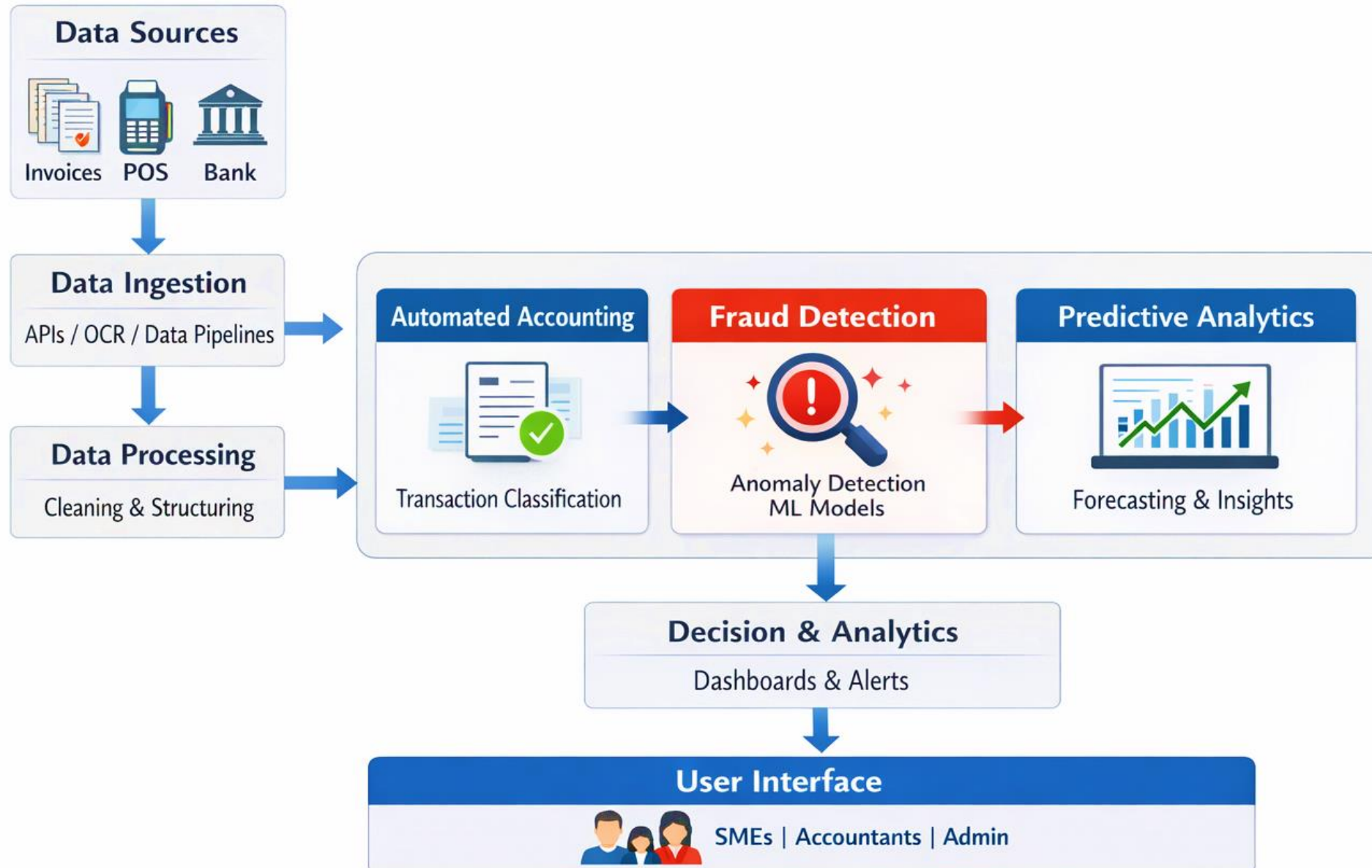
- **Architecture**

- **Data Collection**
- **Data Preprocessing**
- **Automation**
- **Fraud Detection**
- **Explain ability (XAI)**

Architectural plan for the Model construction steps

AI-Based Smart Accounting System

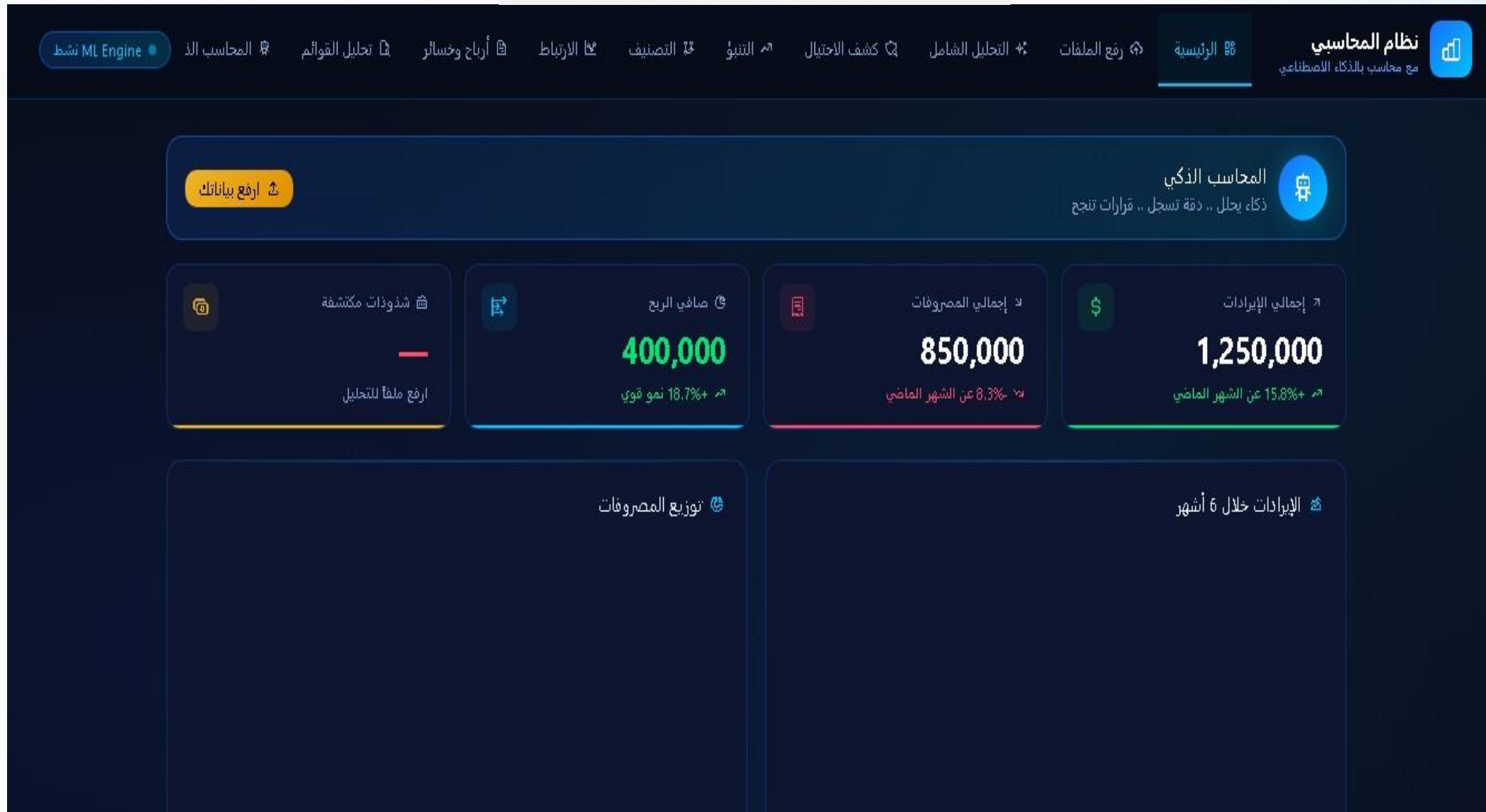
Automate Accounting & Fraud Detection in SMEs





Algorithms & Experimental Results

End-User Interface



The artificial intelligence algorithms used in building the Model.

- 
1. **Chi-Square / SelectKBest**
 2. **SMOTE**
 3. **Random Forest**
 4. **Decision Tree**
 5. **SVM – Support Vector Machine**
 6. **Isolation Forest**
 7. **XGBoost**
 8. **SHAP/TreeExplainer**

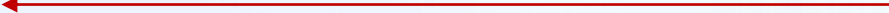
Dataset Description

- “Artificial Intelligence and Accounting Profession” dataset (Mendeley)
<https://data.mendeley.com/datasets/p32vrn9dtz/1>
- Supplemented with synthetic SME data
- Characteristics: class imbalance, tabular financial data

It contains 5,000 financial Records and 33 columns
The columns were classified into 8 main Features:
Transaction_ID, Transaction time, Amount
, Merchant_Description, Account_Type, Previous_Amount, Is_Recurring, Risk level.



Model workflow steps:



DataSet



Preprocessing + Label Encoding



Feature Selection (Chi-Square)



SMOTE To address the imbalance



Classification models

→ Random Forest

→ Decision Tree

→ SVM



Fraud Detection

→ Isolation Forest

→ XGBoost

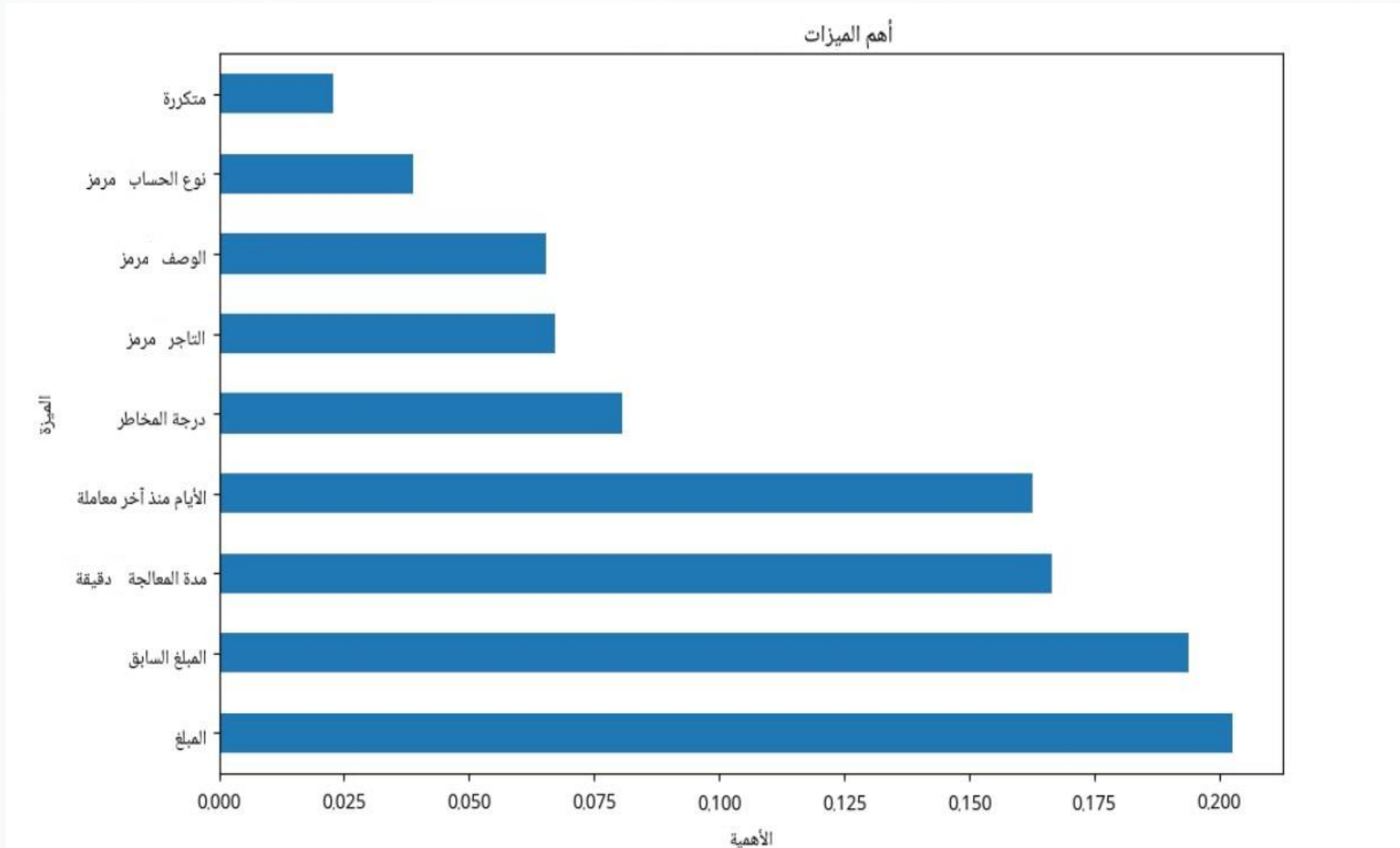


Explainable AI

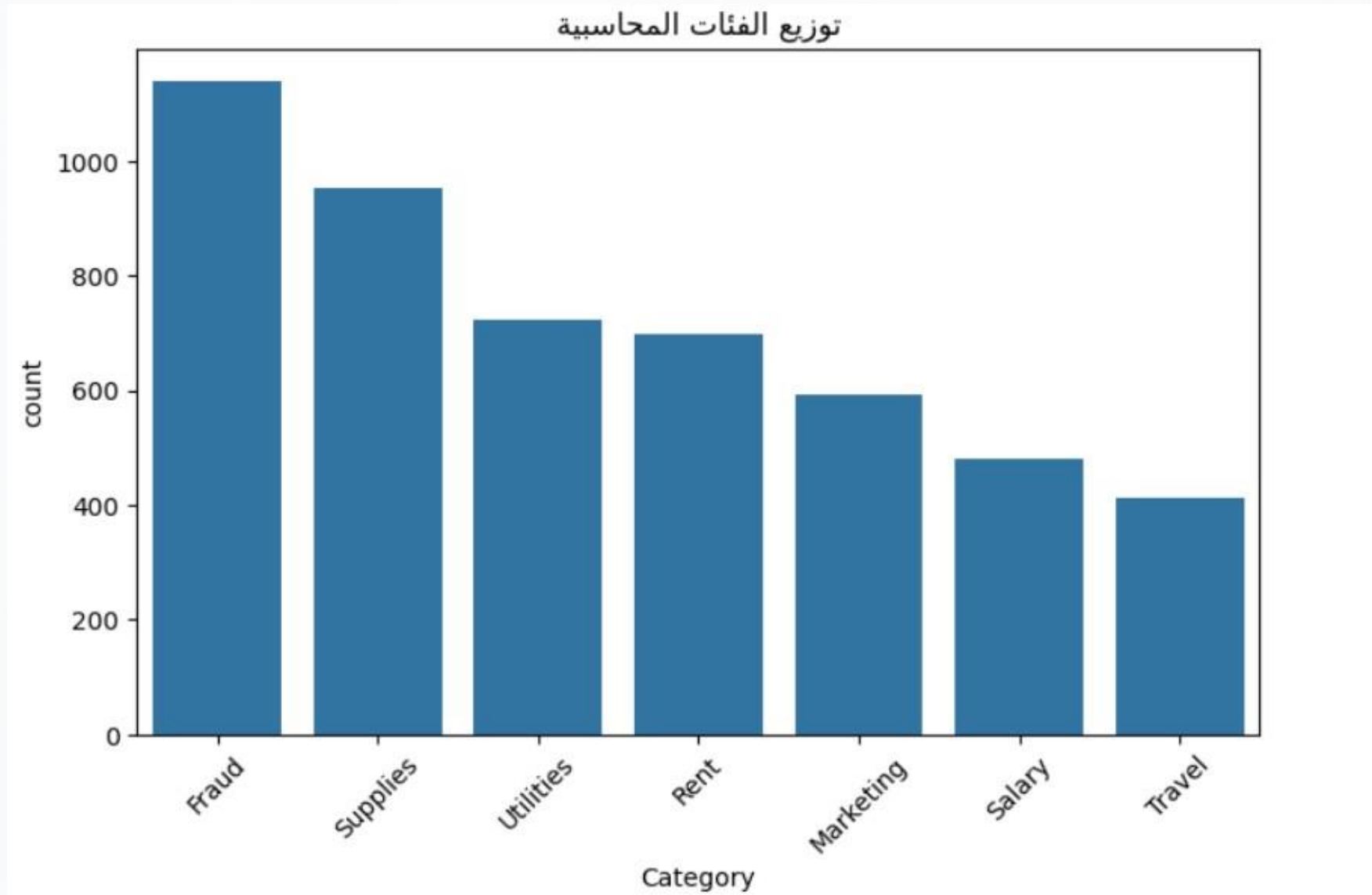
→ SHAP

Relative feature importance ranking extracted from the dataset

Chi-Square /Select K Best Algorithm



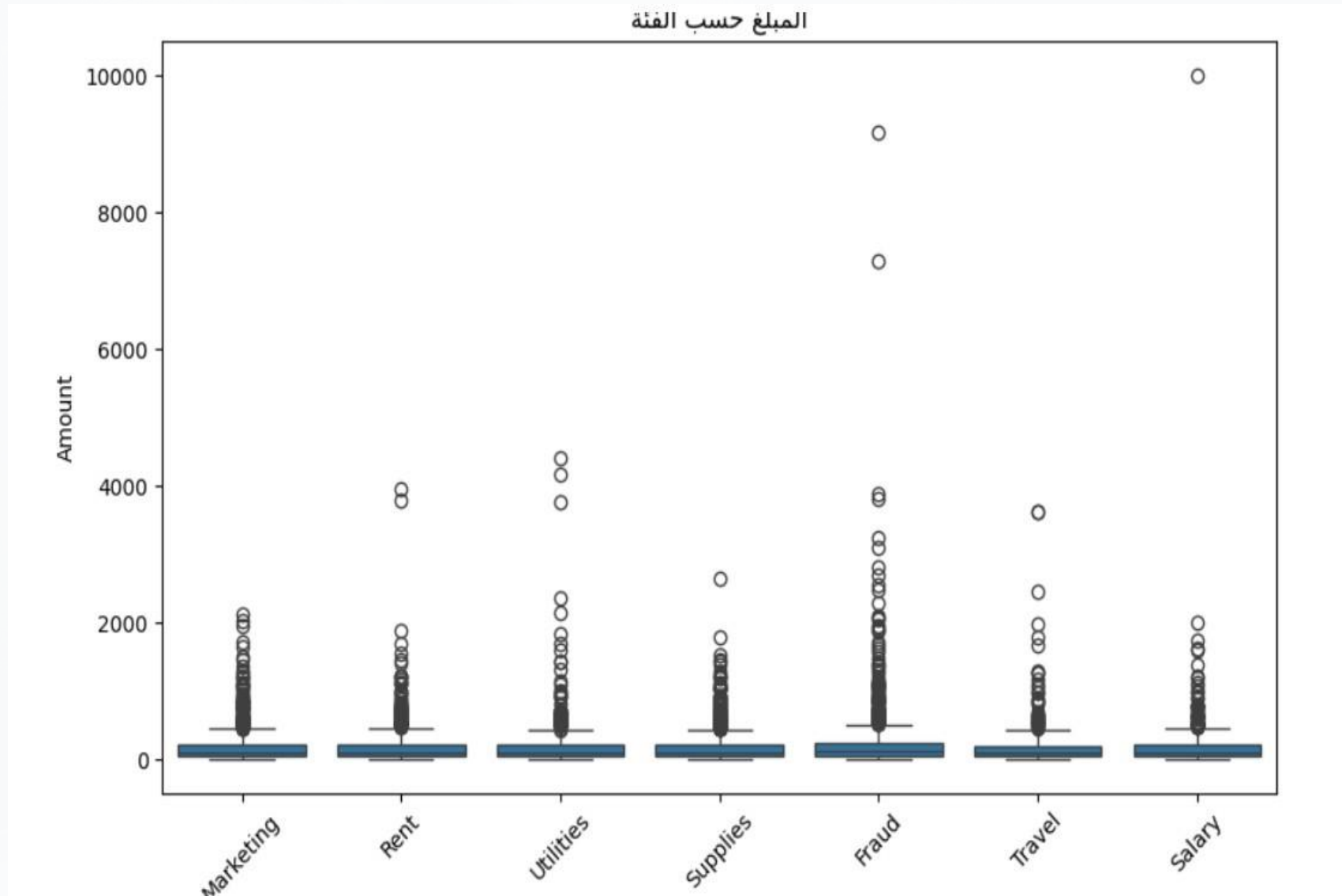
Transaction volume distribution across accounting categories



Random Forest Acuracy:88.47% ,Decision Tree Acuracy:88.6%

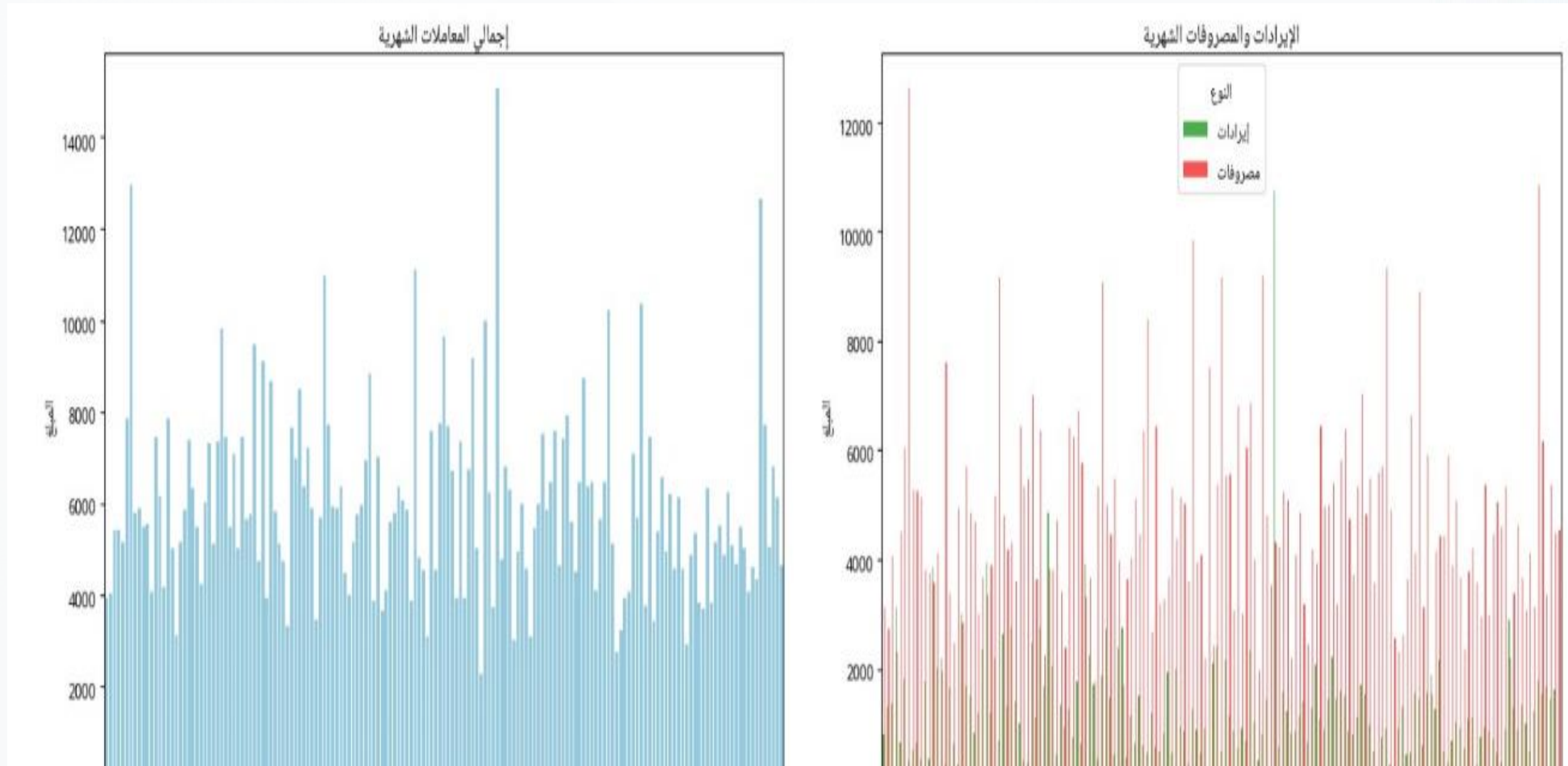
Financial Outlier Analysis by Category

Decision Tree, Random Forest Algorithm



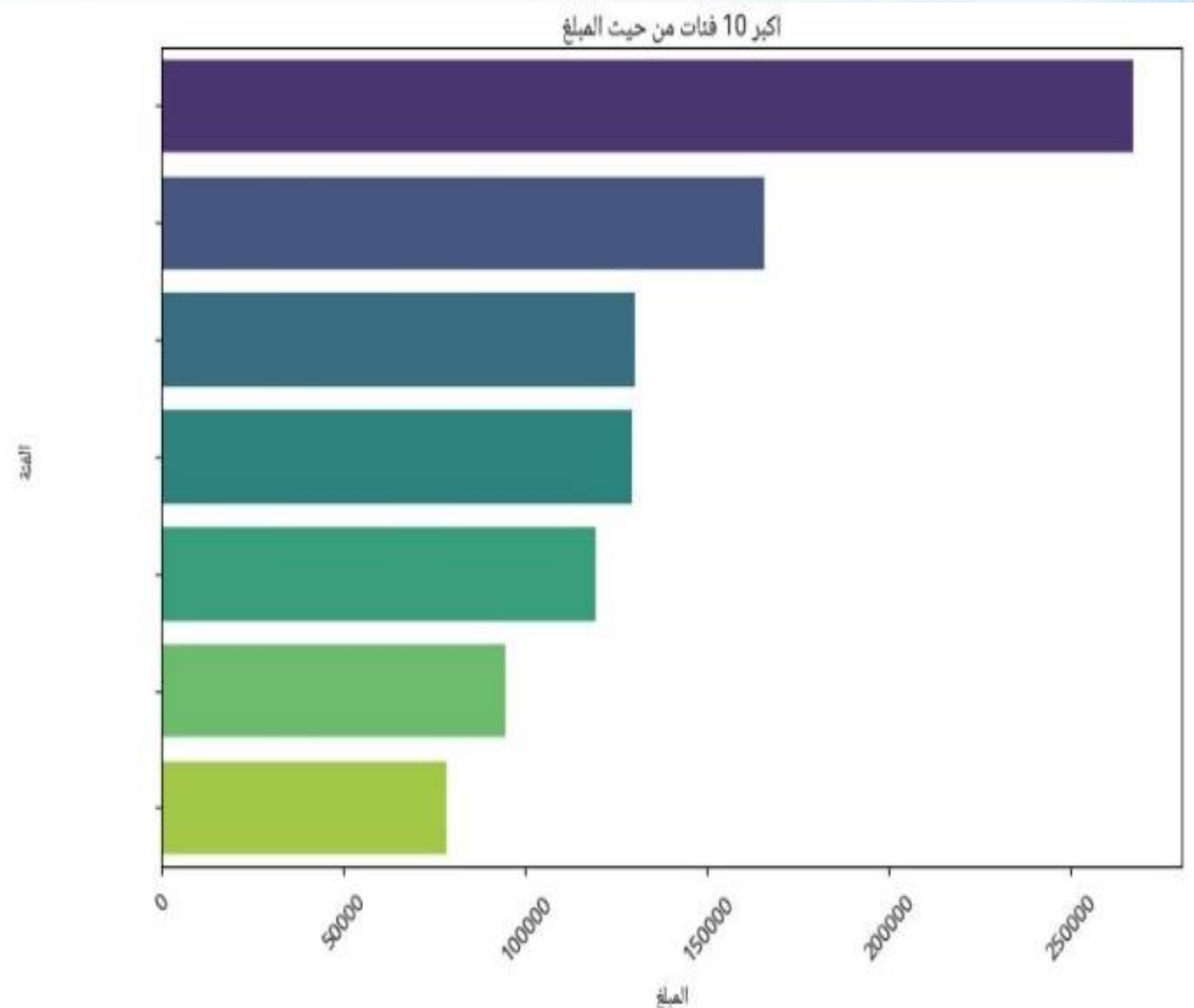
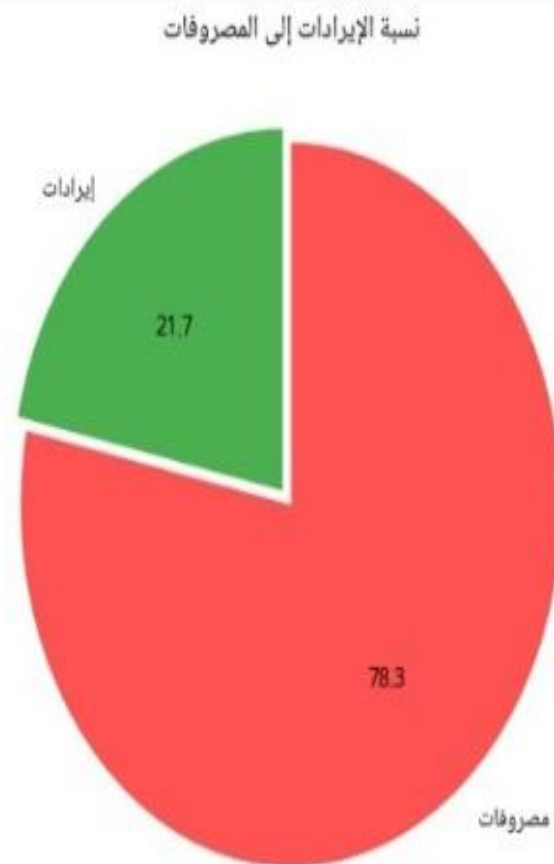
Temporal Transaction Trends and Monthly Revenue Analysis

SVM – Support Vector Machine Algorithm



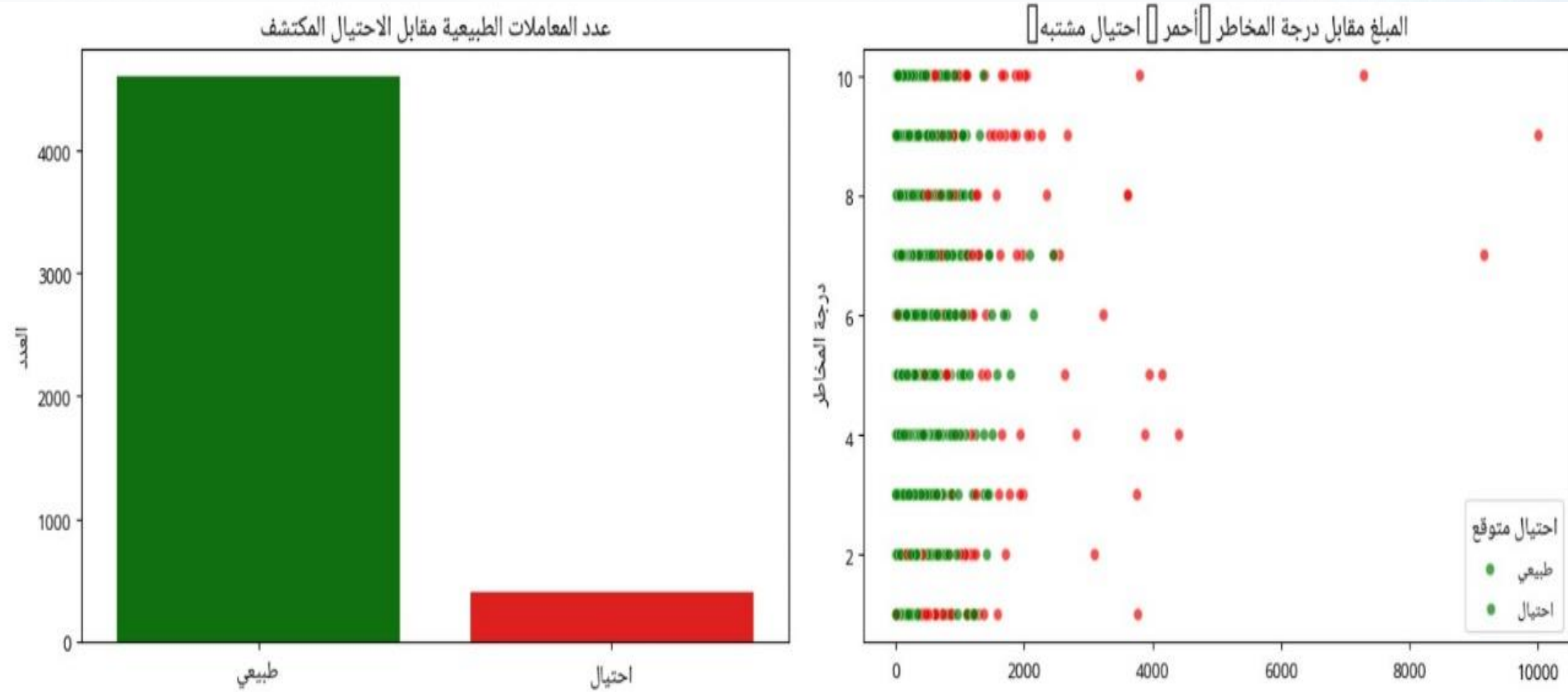
Revenue-to-Expense Ratios & Financial Statements Analysis

SVM – Support Vector Machine Algorithm



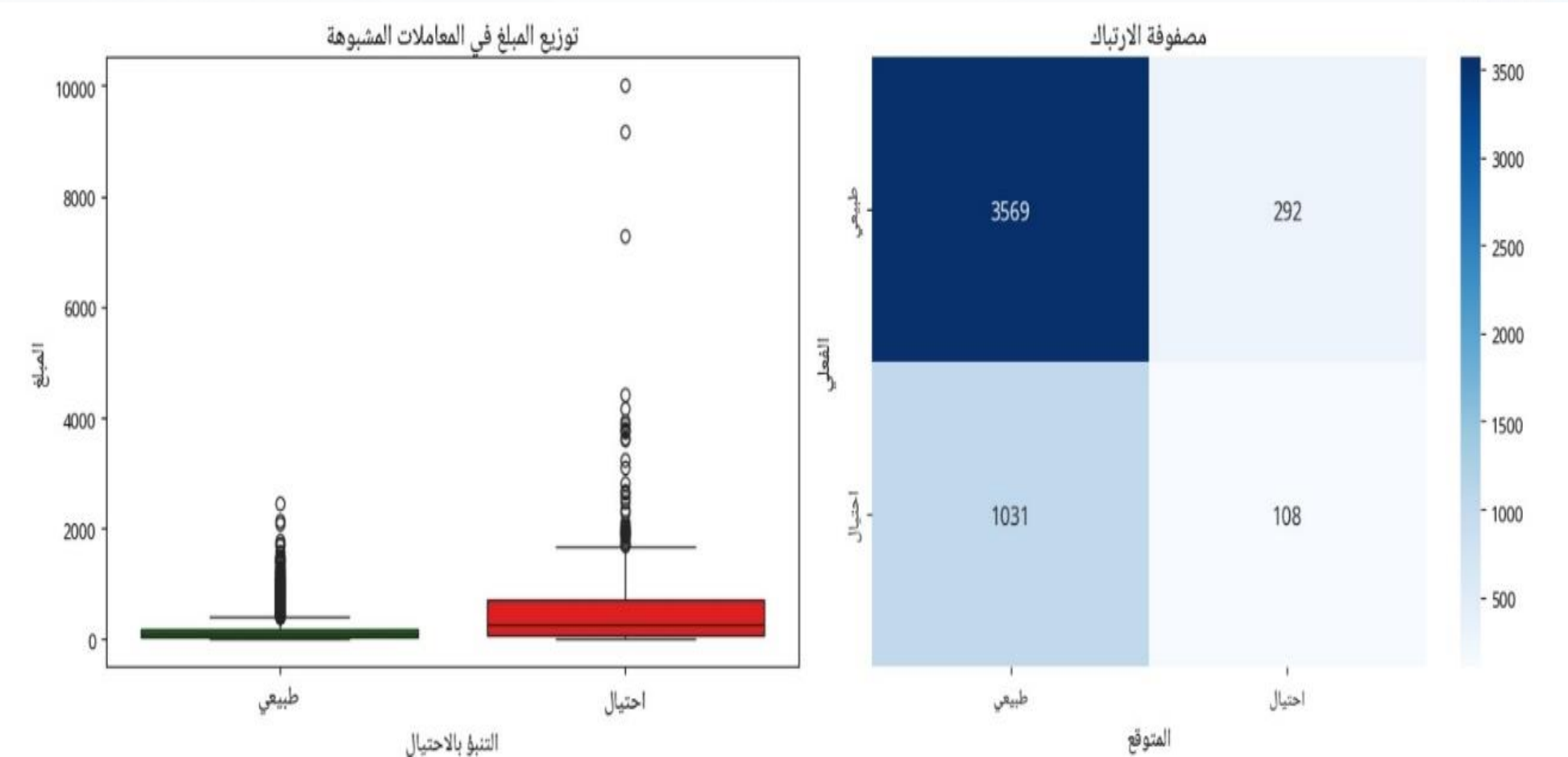
Transaction breakdown between Normal and Fraud (Left)
and Scatter plot of Transaction Amount vs. Risk Score (Right).

Isolation Forest Algorithms



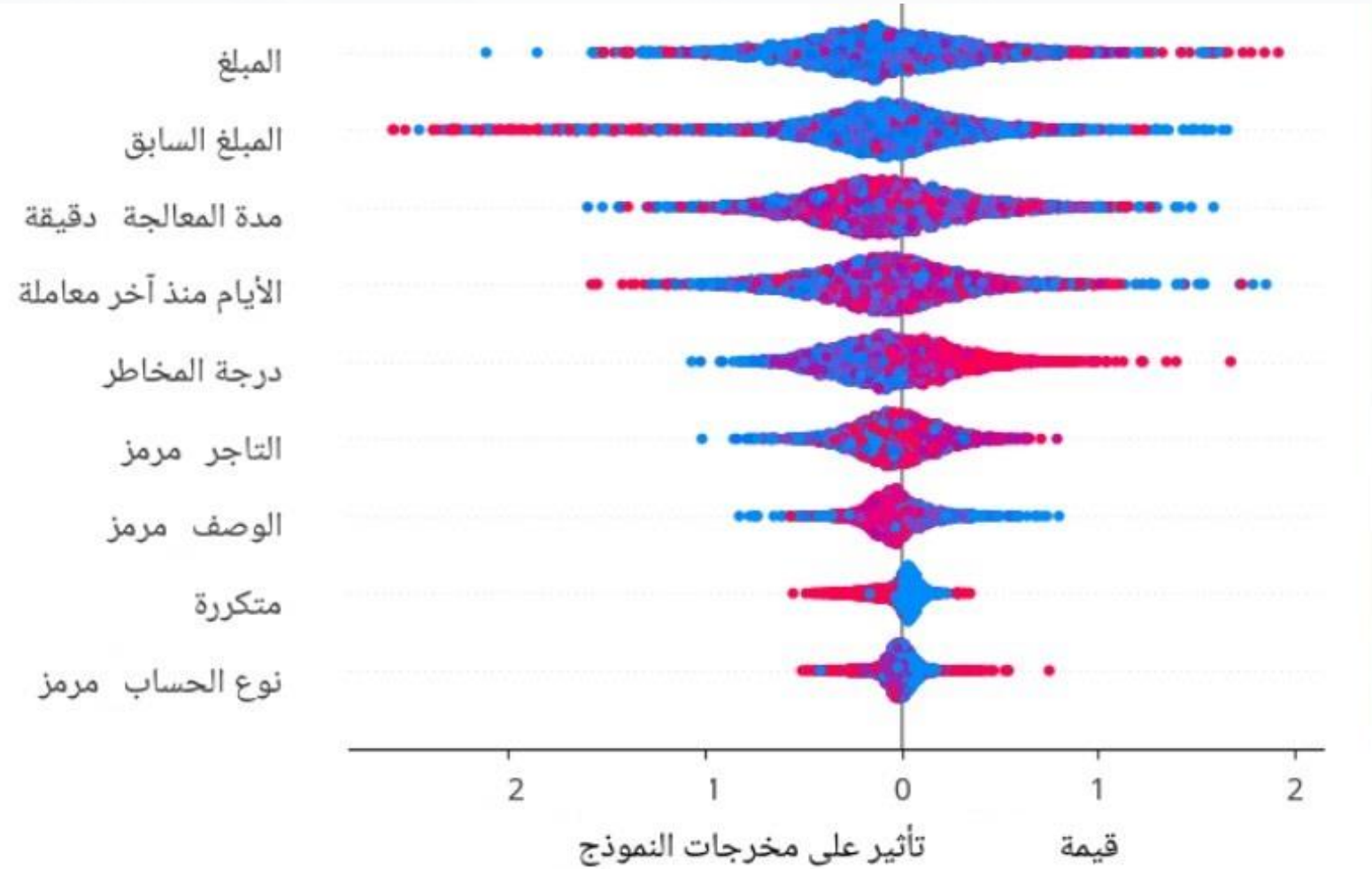
Suspicious transaction amount distribution by predicted class (Left) and Confusion Matrix (Right).

Isolation Forest Algorithm



SHAP Bee swarm plot showing global feature impact and value distributions on model predictions.

SHAP /Tree Explainer Algorithm



تفسير 5 معاملات احتيال

Fraud Detection Performance Report and Metrics Summary

نتائج اكتشاف الاحتيال:

عدد الحالات المكتشفة كاحتيال: 400

تقرير الأداء:

	precision	recall	f1-score	support
Normal	0.78	0.92	0.84	3861
Fraud	0.27	0.09	0.14	1139
accuracy			0.74	5000
macro avg	0.52	0.51	0.49	5000
weighted avg	0.66	0.74	0.68	5000

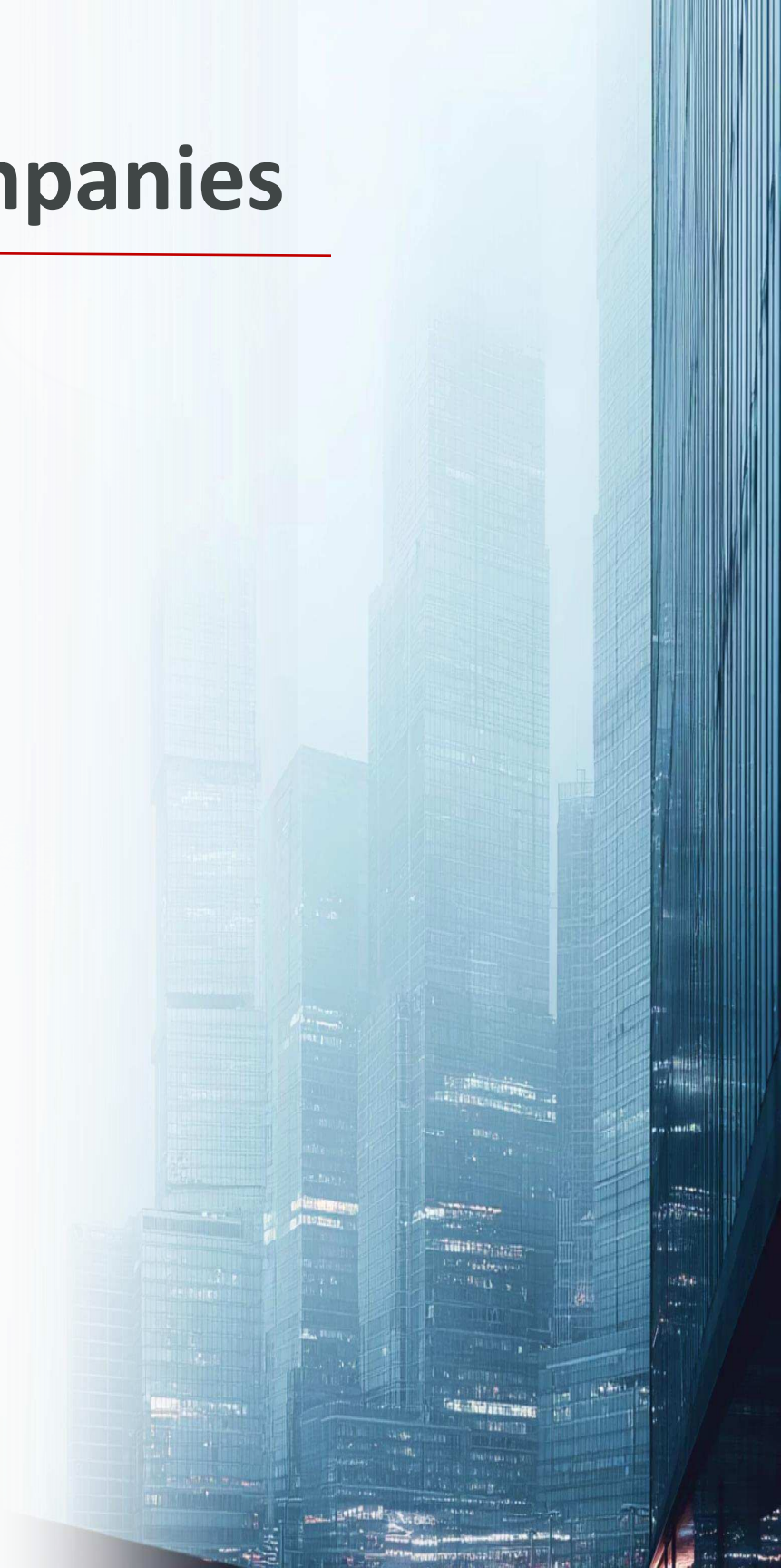
Artificial Intelligence Code in The project.

Smart_Account.ipynb - Colab

[illegible][illegible][illegible]

Benefits of the Model and Added Value for Companies

- **Reducing manual workload:** Automating invoice classification and accounting entries by up to 90%.
- **Enhancing accuracy and compliance:** Directly minimizing human error and adhering to accounting standards.
- **Early Fraud Detection:** Protecting financial assets through real-time alerts prior to fund transfers.
- **Enhanced financial and investment insights:** supporting decision-making through a dashboard and accurate cash flow forecasts.



The Model future vision

ACCOUNTING SYSTEM with AI ACCOUNTANT

Smart Analytics... Precise Recording... Decisive Action

Integrated system for mastering your accounts and making confident decisions.

SMART ACCOUNTANT
Understand and analyze your numbers to provide the best decisions for your business growth.

Profit Increase

Time & Effort

Data Accuracy

Continuous support

Smart Real-time Reports

Thinking AI Accountant

Smart Alerts & Suggestions

Access from Anywhere

Ultimate Safety & Protection

Integrated Management

Ease of Use

Integration & Flexibility

AI-Powered Decisions

YOUR SMART STEP TOWARDS BETTER FINANCIAL MANAGEMENT

Dashboard Sales Purchases Inventory Customers Suppliers Banks Reports Settings

Home

Total Revenue 1,250,000 +15.8%

Total Expenses 850,000 -5.2%

Net Profit 400,000 +18.7%

Cash Balance 320,000 +12.5%

Revenue Over 6 Months

Jan Feb Mar Apr May Jun

Expense Distribution

- Operating (45%)
- Administrative (55%)
- Marketing (30%)
- Other (10%)

AI Accountant Recommendations

- Noticed a 20% increase in marketing expenses from last month.
- There are 3 pending invoices to pay in the coming days.
- Opportunity to reduce administrative costs by 15%.

AI recommendations, as follows, will be analyzed.

View All Recommendations

Acknowledgements



We extend our sincere thanks to Dr. Amjad Qutaish, whose support played a pivotal role in the success and completion of this project. We also express our gratitude to the jury and examination committee, as well as the entire staff at MedOcean University.


```

}

results = []

for name, model in models.items():
    model.fit(X_train_test, y_train_test)
    y_pred = model.predict(X_test)

    print(f'({name}) ({name}) ({name}) ({name})')
    print(classification_report(y_test, y_pred, target_names=labels['Category'], classes=))

    results.append({
        'Model': name,
        'Accuracy': accuracy_score(y_test, y_pred)
    })

# Model Accuracy
results_df = pd.DataFrame(results)
print(results_df)
print(results_df)

===== Random Forest =====
precision    recall    f1-score   support

 Fraud      0.35      0.27      0.30      252
 Marketing  0.12      0.12      0.12      172
 Rent       0.11      0.11      0.11      283
 Salary     0.80      0.80      0.80      186
 Supplies   0.28      0.22      0.25      187
 Travel     0.86      0.88      0.87      117
 Utilities  0.28      0.18      0.19      207

 accuracy   0.38      0.38      0.38      1580
 macro avg  0.15      0.15      0.15      1580
 weighted avg 0.18      0.18      0.18      1580

===== Decision Tree =====
precision    recall    f1-score   support

 Fraud      0.26      0.28      0.23      252
 Marketing  0.14      0.17      0.15      172
 Rent       0.13      0.13      0.13      283
 Salary     0.80      0.18      0.30      186
 Supplies   0.28      0.19      0.23      187
 Travel     0.87      0.18      0.40      117
 Utilities  0.15      0.15      0.15      207

 accuracy   0.36      0.36      0.36      1580
 macro avg  0.15      0.15      0.15      1580
 weighted avg 0.18      0.18      0.17      1580

===== SVM =====
precision    recall    f1-score   support

 Fraud      0.35      0.85      0.48      252
 Marketing  0.12      0.15      0.13      172
 Rent       0.12      0.15      0.13      283
 Salary     0.18      0.15      0.13      186
 Supplies   0.80      0.80      0.80      187
 Travel     0.86      0.14      0.31      117
 Utilities  0.15      0.85      0.30      207

 accuracy   0.38      0.38      0.38      1580
 macro avg  0.14      0.13      0.18      1580
 weighted avg 0.18      0.18      0.40      1580

```

```

# Model Accuracy
0 Random Forest 0.380000
1 Decision Tree 0.360000
2 SVM 0.380000

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score, roc_auc_score
import warnings
warnings.filterwarnings('ignore')

# Load data
df = pd.read_csv('content/sample_data/Marketing.csv')

print(df.info())
print(df.describe())
print(df['Category'].value_counts())

# Data Preprocessing
df['Merchant'] = df['Merchant'].fillna('Unknown', inplace=True)
df['Description'] = df['Description'].fillna('Unknown', inplace=True)

# Label Encoder
le = LabelEncoder()
df['Merchant_enc'] = le.fit_transform(df['Merchant'])
df['Description_enc'] = le.fit_transform(df['Description'])
df['Account_type_enc'] = le.fit_transform(df['Account_type'])

# Feature Engineering
features = ['Amount', 'Processing_time_min', 'Risk_Score',
           'Days_since_last', 'Is_Recurring', 'Previous_Amount',
           'Merchant_enc', 'Description_enc', 'Account_type_enc']

X = df[features]

# Split Data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, df['Account_type_enc'], test_size=0.2, random_state=42)

# Model Training
model = RandomForestClassifier()
model.fit(X_train, y_train)

# Model Evaluation
y_pred = model.predict(X_test)

print(classification_report(y_test, y_pred, target_names=labels['Category'], classes=))

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(10, 10))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.show()

```

```

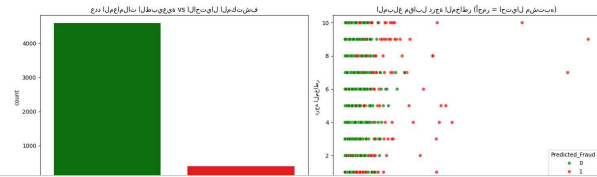
# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(10, 10))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.show()

```

```

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(10, 10))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.show()

```



```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score, roc_auc_score
import warnings
warnings.filterwarnings('ignore')

# Load data
df = pd.read_csv('content/sample_data/Marketing.csv')

print(df.info())
print(df.describe())
print(df['Category'].value_counts())

# Data Preprocessing
df['Merchant'] = df['Merchant'].fillna('Unknown', inplace=True)
df['Description'] = df['Description'].fillna('Unknown', inplace=True)

# Label Encoder
le = LabelEncoder()
df['Merchant_enc'] = le.fit_transform(df['Merchant'])
df['Description_enc'] = le.fit_transform(df['Description'])
df['Account_type_enc'] = le.fit_transform(df['Account_type'])

# Feature Engineering
features = ['Amount', 'Processing_time_min', 'Risk_Score', 'Days_since_last',
           'Is_Recurring', 'Previous_Amount', 'Merchant_enc',
           'Description_enc', 'Account_type_enc']

X = df[features]

# Split Data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, df['Account_type_enc'], test_size=0.2, random_state=42)

# Model Training
model = RandomForestClassifier()
model.fit(X_train, y_train)

# Model Evaluation
y_pred = model.predict(X_test)

print(classification_report(y_test, y_pred, target_names=labels['Category'], classes=))

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(10, 10))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.show()

```

```
plt.title("الخرات جي ڦر")
plt.tight_layout()
plt.show()

# ===== 5. ڪم ڪندڙ ڏيکاري سگهڻ =====
# ڦر ڏيکاري 5 ڪم ڪندڙ ڏيکاري سگهڻ
fraud_indices = df[df['Category'] == 'Fraud'].index[1:]

print("ڪم ڪندڙ ڏيکاري سگهڻ")
for i, idx in enumerate(fraud_indices):
    print(f"--- ڪم ڪندڙ ڏيکاري سگهڻ (Transaction_ID: {df.loc[idx, 'Transaction_ID']}) ---")
    print(f"ڦر: {df.loc[idx, 'Amount'] | Risk Score: {df.loc[idx, 'Risk_Score']}")

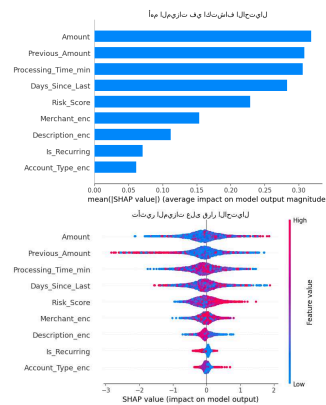
# Force Plot
shap.force_plot(explainer.expected_value,
                shap_values[idx],
                X.loc[idx],
                matplotlib.rcParams.update({'font.size': 14}))

plt.title(f"ڦر ڏيکاري سگهڻ (i={i})")
plt.show()
```

ڪم ڪندڙ ڏيکاري سگهڻ

ڦر ڏيکاري سگهڻ

	precision	recall	f1-score	support
0	0.75	0.86	0.80	1210
1	0.26	0.05	0.08	342
accuracy			0.75	1280
macro avg	0.52	0.58	0.47	1280
weighted avg	0.66	0.75	0.68	1280



ڪم ڪندڙ ڏيکاري سگهڻ

--- 1 Use (Transaction_ID: 7000000) ---

11.11: ڦر | Risk Score: 9



--- 2 Use (Transaction_ID: 7000000) ---

11.11: ڦر | Risk Score: 9



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import random as rnd
import warnings
warnings.filterwarnings("ignore")

# ===== 1. ڊيٽا ڏيکاري سگهڻ =====
df = pd.read_csv("../content/sample_data/Weekend.csv")

# ڊيٽا ڏيکاري سگهڻ
df['Merchant'].fillna('Unknown', inplace=True)
df['Description'].fillna('Unknown', inplace=True)

# ڊيٽا ڏيکاري سگهڻ
from sklearn.preprocessing import LabelEncoder
le_merchant = {}
for col in ['Merchant', 'Description', 'Account_Type']:
    le = LabelEncoder()
    df[col + '_enc'] = le.fit_transform(df[col])
    le_dict[col] = le

# ڊيٽا ڏيکاري سگهڻ
features = ['Amount', 'Processing_Time_min', 'Risk_Score', 'Days_Since_Last',
            'Is_Recurring', 'Previous_Amount', 'Merchant_enc',
            'Description_enc', 'Account_Type_enc']

X = df[features]
y = df['Category'] # 0 = Fraud, 1 = Normal

print(f"ڊيٽا ڏيکاري سگهڻ: {X.shape[0]}")

# ===== 2. ڊيٽا ڏيکاري سگهڻ =====
from sklearn import train_test_split
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42, stratify=y)

model = SGDClassifier(loss='log_loss', learning_rate=0.1, max_depth=3, random_state=42)
model.fit(X_train, y_train)

# ڊيٽا ڏيکاري سگهڻ
from sklearn.metrics import classification_report
print(f"ڊيٽا ڏيکاري سگهڻ")
print(classification_report(y_test, model.predict(X_test)))

# ===== 3. ڊيٽا ڏيکاري سگهڻ =====
import shap

# ڊيٽا ڏيکاري سگهڻ
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X_test)

# ڊيٽا ڏيکاري سگهڻ
plt.figure(figsize=(15, 10))

# 1. ڊيٽا ڏيکاري سگهڻ (Summary Plot)
shap.summary_plot(shap_values, X_test, plot_type="bar", show=False)
plt.title("ڊيٽا ڏيکاري سگهڻ")
plt.tight_layout()
plt.show()

# 2. ڊيٽا ڏيکاري سگهڻ (Force Plot)
shap.summary_plot(shap_values, X_test, show=False)
plt.title("ڊيٽا ڏيکاري سگهڻ")
plt.tight_layout()
plt.show()

# ===== 5. ڪم ڪندڙ ڏيکاري سگهڻ =====
# ڦر ڏيکاري 5 ڪم ڪندڙ ڏيکاري سگهڻ
fraud_indices = df[df['Category'] == 'Fraud'].index[1:]

print("ڪم ڪندڙ ڏيکاري سگهڻ")
for i, idx in enumerate(fraud_indices):
    print(f"--- ڪم ڪندڙ ڏيکاري سگهڻ (Transaction_ID: {df.loc[idx, 'Transaction_ID']}) ---")
    print(f"ڦر: {df.loc[idx, 'Amount'] | Risk Score: {df.loc[idx, 'Risk_Score']}")

# Force Plot
shap.force_plot(explainer.expected_value,
                shap_values[idx],
                X.loc[idx],
                matplotlib.rcParams.update({'font.size': 14}))

plt.title(f"ڦر ڏيکاري سگهڻ (i={i})")
plt.show()
```

