

حل الطلب الثالث:

مقدمة:

الهدف هو استخدام نموذج شجرة القرار (Decision Tree) لتحديد أهمية الخصائص المؤثرة على اختيار الإجراء التالي للعميل، ومن ثم استخدام هذا النموذج لتقديم أفضل أربعة أنواع من الإجراءات (Top 4 Actions) مثل (Email) ، (Call) ، (Meeting)

البرامج المستخدمة:

استخدمت (Google Colab (Jupyter Notebook).

المكتبات المستخدمة:

- Pandas
- Scikit-learn
- Matplotlib

قراءة البيانات

قمت بقراءة ملف الإكسل النظيف وتحويله إلى DataFrame باستخدام مكتبة Pandas

```
import pandas as pd
df = pd.read_excel("data_clean.xlsx")
```

تجهيز البيانات للنمذجة

تم التعامل مع القيم المفقودة في الأعمدة المهمة مثل نوع الإجراء ومرحلة الفرصة ومصدر النظام، كما تم تحويل عمود is_lead إلى قيم رقمية.

```
for col in ['types', 'opportunity_stage', 'SourceSystem']:
    df[col] = df[col].fillna('unknown')

df['is_lead'] = pd.to_numeric(df['is_lead'], errors='coerce')
df['is_lead'] = df['is_lead'].fillna(0).astype(int)
```

ترميز البيانات النصية

بما أن نموذج شجرة القرار لا يتعامل مع البيانات النصية، تم استخدام Ordinal Encoder لتحويل الأعمدة النصية إلى قيم رقمية.

```
encoder = OrdinalEncoder(handle_unknown='use_encoded_value', unknown_value=-1)
encoder.fit(df[['types', 'opportunity_stage']])
df[['types_encoded', 'stage_encoded']] = encoder.transform(df[['types', 'opportunity_stage']])
```

بناء وتدريب نموذج شجرة القرار

تم بناء نموذج Decision Tree باستخدام نوع الإجراء وحالة العميل كمدخلات، ومرحلة الفرصة كمخرج.

```
# تدريب نموذج شجرة القرار
x = df[['types_encoded', 'is_lead']]
y = df['stage_encoded']
clf = DecisionTreeClassifier(max_depth=4)
clf.fit(x, y)
```

عرض شجرة القرار

تم عرض شجرة القرار لفهم كيفية اتخاذ النموذج للقرارات.

```
plt.figure(figsize=(20, 10))
plot_tree(clf, feature_names=['types_encoded', 'is_lead'], filled=True, fontsize=12)
plt.title("Decision Tree for Action Prediction")
plt.tight_layout()
plt.show()
```

ومن ثم تم إنشاء دالة تقوم بالتنبؤ بالمرحلة التالية للفرصة بناءً على نوع الإجراء وحالة العميل.

```
# دالة التنبؤ بالمرحلة التالية
def predict_next_stage(action_type, is_lead):
    encoded = encoder.transform([[action_type, 'unknown']])
    action_encoded = encoded[0][0]
```