

# Classification Models Performance Evaluation

## تقييم اداء نماذج التصنيف

يعرض هذا القسم نتائج تقييم نماذج التصنيف المستخدمة ومقارنتها من حيث الأداء

# KNN Hyperparameter Tuning Result

```
print("Best Score:", KNN_grid_search.best_score_)  
print("Best Parameters:", KNN_grid_search.best_params_)
```

Fitting 3 folds for each of 3 candidates, totalling 9 fits

Best Score: 0.8211140503217799

Best Parameters: {'kneighborsclassifier\_\_metric': 'manhattan', 'kneighborsclassifier\_\_n\_neighbors': 80, 'kneighborsclassifier\_\_weights': 'distance'}

استخدمنا GridSearch مع GroupKFold عشان نختار افضل بارميتر لل KNN مع تجنب DataLeakge  
افضل نتيجة بعد عمل تونينغ على بيانات التدريب كانت **F1-score=0.82**  
تعد هذه النتيجة جيدة مقارنة ببساطة لموديل وحساسيته مع الداتا الكبيرة منقدر نخليها اعلى لل اكيورسي بس بصير عنا  
**DataLeakge**

# KNN Final Model Result

```
[ ]
... Best Threshold: 0.35
      precision    recall  f1-score   support

         0       0.75       0.73       0.74      42675
         1       0.81       0.83       0.82      60380

 accuracy                   0.78      103055
 macro avg       0.78       0.78       0.78      103055
 weighted avg    0.78       0.78       0.78      103055

KNN Accuracy 0.7848333414196303

... 'k=80 got accuracy of 0.9123456790123457 > k=30 '
```

استخدمنا هون Threshold نظرا لحساسية نموذج KNN وكانت افضل قيمة لاله =0.35 وهاد الاشئ ادى الى تحسين اداء النموذج وتحقيق

توازن افضل بين precision&recall

**KNN Accuracy=0.78 ❖**

**precision=0.81 ❖**

**recall=0.83 ❖**

# Logistic Regression Hyperparameter Tuning Result

```
print("\nBest Score:", LoR_grid_search.best_score_)  
print("Best Parameters:", LoR_grid_search.best_params_)
```

[ ]

```
... Fitting 3 folds for each of 10 candidates, totalling 30 fits  
C:\Users\owndi\AppData\Roaming\Python\Python313\site-packages\sklearn\linear\_model\logistic.py:1135:  
warnings.warn(  

```

```
Best Score: 0.791938732224401
```

```
Best Parameters: {'logisticregression__C': 0.1, 'logisticregression__penalty': 'l2'}
```

تم استخدام GridSearch مع GroupKFold لأختيار أفضل باراميترز لل LoR حيث حقق النموذج افضل اداء عند :

**F1-score=0.79 ❖**

**Penalty=l2 ❖**

**C=0.1 ❖**

# Logistic Regression Final Model Result

```
C:\Users\owndi\AppData\Roaming\Python\Python313\site-packages\sklearn\linear_model\logis
warnings.warn(
--- Best Threshold Used: 0.3 ---
      precision    recall  f1-score   support

     0       0.73      0.89      0.80     42675
     1       0.91      0.76      0.83     60380

 accuracy          0.82     103055
 macro avg         0.82      0.83      0.81     103055
weighted avg         0.83      0.82      0.82     103055

Logistic Accuracy: 0.8151
```

تم استخدام Threshold لأن النموذج يعتمد على الاحتمالات مما ساعد على تحسين دقة لموديل بالإضافة حققنا توازن افضل بين precision&recall

**Best Threshold =0.3 ❖**

**Logistic Acc=0.81 ❖**

**Precision=0.91 ❖**

**Recall=0.76 ❖**

# Random Forest Hyperparameter Tuning Result& Final Result

```
print("\nBest Score:", RF_grid_search.best_score_)  
print("Best Parameters:", RF_grid_search.best_params_)
```

Python

```
Best Score: 0.8489607431928764
```

```
Best Parameters: {'randomforestclassifier__class_weight': 'balanced_subsample', 'randomforestclassifier__max_depth': 10, 'randomf
```

تم استخدام GridSearch مع GroupKFold لأختيار أفضل باراميترز لل Random Forest حيث حقق النموذج افضل اداء عند :

**F1-score:0.84 ❖**

**RF-estimators: 200 ❖**

**RF-max-depth:10 ❖**

**RF-weight :‘balance’ ❖**

**RF-min-samples-leaf:10 ❖**

**RF-min-samples-split : 30 ❖**

**ACC Final Result: 0.76**

# Linear Support Vector Machine

```
... SVM with Linear Kernel Accuracy: 0.8422420218217954
      precision    recall  f1-score   support

         0         0.85        0.74        0.79        300383
         1         0.84        0.91        0.87        436220

 accuracy                   0.84        736603
 macro avg                   0.84        0.83        0.83        736603
 weighted avg                0.84        0.84        0.84        736603

... 'The best Score C is 1.0 as it gives the highest accuracy of 0.8091'
```

استخدمنا هون الموديل SVM مع Linear Kernal

**ACC SVM with kernal=0.84 ❖**

**Precision=0.84 ❖**

**Recall=0.91 ❖**

The best Score C is 1.0 as it gives the highest accuracy of 0.8091

# Linear Support Vector Machine

```
... SVM with Linear Kernel Accuracy: 0.8422420218217954
      precision    recall  f1-score   support

         0         0.85        0.74        0.79        300383
         1         0.84        0.91        0.87        436220

 accuracy
macro avg         0.84        0.83        0.83        736603
weighted avg         0.84        0.84        0.84        736603

... 'The best Score C is 1.0 as it gives the highest accuracy of 0.8091'
```

جربنا linear SVM مع اكثر من قيمة لل parameter C عشان نشوف تأثيرها على الاداء:

• SVM Score with Linear Kernel Accuracy: 0.7996820567076329 -> with c= 10

• SVM Score with Linear Kernel Accuracy: 0.8020444783838858 -> with c= 0.1

• SVM Score with Linear Kernel Accuracy: 0.7984317144654071 with c=100

اعلى Accuracy طلعت عند C=1 مقارنة بباقي القيم ف اعتبرناها افضل اختيار :

**SVM Score with Linear Kernel Accuracy: 0.809161776248863 -> with c= 1.0**



# Decision Tree Hyperparameter Tuning Result

```
...
      precision    recall  f1-score   support

     0       0.69       1.00       0.82     42675
     1       1.00       0.69       0.82     60380

 accuracy                   0.82     103055
 macro avg       0.85       0.84       0.82     103055
 weighted avg    0.87       0.82       0.82     103055
```

Decision Tree Accuracy: 0.8172

```
... 'Decision Tree Accuracy: 0.8172 with max_depth=5, min_samples_leaf=5, min_samples_split=5 '
```

استخدمنا Decision Tree مع مجموعة من Numerical Features حيث حقق النموذج افضل اداء عند:

**Decision Tree Accuracy: 0.81 ❖**

**max\_depth=5 ❖**

**min\_samples\_leaf=5 ❖**

**min\_samples\_split=5 ❖**

# Decision Tree Final Model Result

```
... Threshold Recall 0 Recall 1 Gap
-----
              precision      recall  f1-score   support

              0      0.68      0.74      0.71      42675
              1      0.80      0.75      0.78      60380

    accuracy                  0.75      103055
  macro avg              0.74      0.75      0.74      103055
weighted avg              0.75      0.75      0.75      103055

Balanced Accuracy: 0.7461
```

تم استخدام مقياس ال Gap (للفرق بين (0,1) recall) لاختيار threshold مناسب بهدف تحقيق توازن بدل الاعتماد على الاكيورسي فقط لانه البيانات غير متوازنة

**Balanced ACC:0.74**

**Precision:0.80**

**Recall:0.75**

# XGBoost Final Model Result

```
0.89      0.3734      0.98      0.23
F1 (بناء على) افضل عتبة: 0.25000000000000006

تقرير الموديل النهائي:
      precision      recall  f1-score      support
0      0.79      0.56      0.65      42675
1      0.74      0.90      0.81      60380

accuracy      0.76      103055
macro avg      0.77      0.73      0.73      103055
weighted avg      0.76      0.76      0.75      103055

الدقة العامة (Accuracy): 0.7556
```

يوضح هاد الشكل نتيجة XGBoost النهائي اخترنا افضل threshold بدل القيمة الافتراضية عشان نعمل بالانس بين recall&precision حققنا افضل اداء:

**ACC:0.75 ❖**

**Threshold:0.25 ❖**

**Precision:0.74 ❖**

**Recall:0.90 ❖**

# XGBoost Final Model Result

```
Balanced Threshold: 0.1000
precision    recall  f1-score   support

0           0.70      1.00      0.82     42675
1           1.00      0.69      0.82     60380

accuracy          0.82     103055
macro avg         0.85      0.85      0.82     103055
weighted avg      0.87      0.82      0.82     103055

XGBoost Accuracy: 0.8193
```

جربنا اكثر من قيمة لل threshold مع انه بتطلع الاكويروسي احسن من يلي قبل لكن ملاحظ انه في عندنا dataLeakge ك مثال :  
استخدمنا Threshold= 0.1

# LightGBM Final Model Result

```
..  Balanced Threshold: 0.1000
      precision    recall  f1-score   support

         0         0.76      0.59      0.67      42675
         1         0.75      0.87      0.80      60380

   accuracy              0.75      103055
  macro avg              0.76      0.73      0.74      103055
 weighted avg              0.75      0.75      0.75      103055

Accuracy: 0.8194
LightGBM Accuracy: 0.8194

..  'LightGBM Accuracy: 0.8255 with n_estimators=300, max_depth=20 , learning_rate=0.1'
```

تم تدريب موديل lightGBM ب استخدام مجموعة معينة features حيث حقق النموذج افضل اداء عند:

**Blance Threshold: 0.1 ❖**

**Accuracy:0.82 ❖**