Rakamin Academy

Uranus

Dokumen Laporan Final Project (Stage 3)

(dipresentasikan setiap sesi mentoring)

Rakamin Academy

Modeling

1A. Split Data Train & Test

Dataset dibagi menjadi 80% untuk training dan 20% untuk testing menggunakan function train_test_split.

```
x_train, x_test, y_train, y_test = train_test_split(df.drop(columns=['Revenue']), df['Revenue'], test_size=0.2, random_state=28)
```

Handling imbalance menggunakan metode SMOTE dengan sampling strategy all.

```
x_smote, y_smote = over_sampling.SMOTE('all', random_state=28).fit_resample(x_train, y_train)
```

```
print('Original')
print(pd.Series(y_train).value_counts())
print('\nSMOTE')
print(pd.Series(y_smote).value_counts())
```

```
Original
0    7561
1    1208
Name: Revenue, dtype: int64

SMOTE
1    7561
0    7561
Name: Revenue, dtype: int64
```

1B. Modeling

Digunakan library pycaret untuk membandingkan performa dari beberapa algoritma yang sesuai dengan data yang digunakan.

```
# initialize setup
s = setup(df,
          target = 'Revenue',
          train_size = 0.8,
          fix_imbalance = True,
          numeric_features = df.drop('Revenue', axis=1).columns.tolist())
```

1C. Model Evaluation: Pemilihan dan perhitungan metrics

```
# compare all models
best = compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lightgbm	Light Gradient Boosting Machine	0.8969	0.9220	0.6342	0.6263	0.6294	0.5696	0.5701	0.492
et	Extra Trees Classifier	0.8953	0.9108	0.6069	0.6255	0.6158	0.5552	0.5555	1.379
rf	Random Forest Classifier	0.8943	0.9153	0.6408	0.6125	0.6258	0.5643	0.5649	1.861
gbc	Gradient Boosting Classifier	0.8930	0.9192	0.7176	0.5942	0.6494	0.5870	0.5911	2.583
ada	Ada Boost Classifier	0.8857	0.9050	0.6953	0.5709	0.6267	0.5600	0.5640	0.730
ridge	Ridge Classifier	0.8789	0.0000	0.7333	0.5467	0.6257	0.5554	0.5644	0.078
lda	Linear Discriminant Analysis	0.8789	0.9055	0.7333	0.5467	0.6257	0.5554	0.5644	0.133
dummy	Dummy Classifier	0.8619	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.070
dt	Decision Tree Classifier	0.8612	0.7433	0.5805	0.4981	0.5360	0.4550	0.4569	0.165
lr	Logistic Regression	0.8523	0.9066	0.7746	0.4792	0.5916	0.5076	0.5298	1.031
knn	K Neighbors Classifier	0.8091	0.8394	0.7349	0.3978	0.5157	0.4099	0.4401	0.675
svm	SVM - Linear Kernel	0.8060	0.0000	0.8348	0.4184	0.5501	0.4474	0.4948	0.209
nb	Naive Bayes	0.5990	0.8421	0.8811	0.2405	0.3778	0.2054	0.3003	0.079
qda	Quadratic Discriminant Analysis	0.1381	0.5000	1.0000	0.1381	0.2427	0.0000	0.0000	0.090

- Tim kami berfokus kepada metric **precision** karena ingin meningkatkan **False Negatif** untuk meningkatkan angka pembelian dari pengunjung agar mendapatkan revenue.
- Melihat metric **AUC** untuk mendapatkan performa model yang tinggi.
- Melihat **waktu eksekusi** untuk mendapatkan waktu yang efektif (rendah) saat membuat model.
- Berdasarkan hasil perbandingan model dengan Pycaret dengan mepertimbangan metrics, model **Light Gradient Boosting Machine** menjadi pilihan karena memiliki nilai **Precision cukup tinggi**, nilai **AUC yang tinggi**, dan **waktu eksekusi yang relatif rendah**, serta **metric-metric lainnya yang lebih baik diantara model lainnya**.

1D. Model Evaluation: Apakah model sudah best-fit?

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.9111	0.9399	0.7541	0.6571	0.7023	0.6503	0.6524
1	0.8951	0.9008	0.6098	0.6303	0.6198	0.5590	0.5591
2	0.8985	0.9261	0.6016	0.6491	0.6245	0.5659	0.5664
3	0.9019	0.9264	0.6098	0.6637	0.6356	0.5791	0.5798
4	0.8962	0.9242	0.6098	0.6356	0.6224	0.5623	0.5625
5	0.8974	0.9227	0.5854	0.6486	0.6154	0.5564	0.5573
6	0.8940	0.9108	0.6504	0.6154	0.6324	0.5705	0.5708
7	0.9019	0.9211	0.6748	0.6434	0.6587	0.6015	0.6017
8	0.9019	0.9334	0.6667	0.6457	0.6560	0.5988	0.5989
9	0.9041	0.9133	0.6393	0.6610	0.6500	0.5945	0.5946
Mean	0.9002	0.9219	0.6402	0.6450	0.6417	0.5838	0.5844
Std	0.0048	0.0107	0.0472	0.0140	0.0249	0.0273	0.0276

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Light Gradient Boosting Machine	0.9106	0.9353	0.6233	0.6926	0.6561	0.6049	0.6061

- Nilai **AUC test sedikit lebih besar dari nilai trainnya** yakni selisih **0.0134.**
- Nilai **precision test lebih besar dari nilai trainnya** yakni selisih **0.0217.**
- Nilai **accuracy test sedikit lebih besar dari nilai trainnya** yaitu selisih **0.0104**.
- Dari pengamatan diatas tim kami menyimpulkan bahwa model ini sudah cukup **Best-Fit**.

1E. Hyperparameter Tuning

Hyperparameter tuning menggunakan parameter boosting_type dan objective

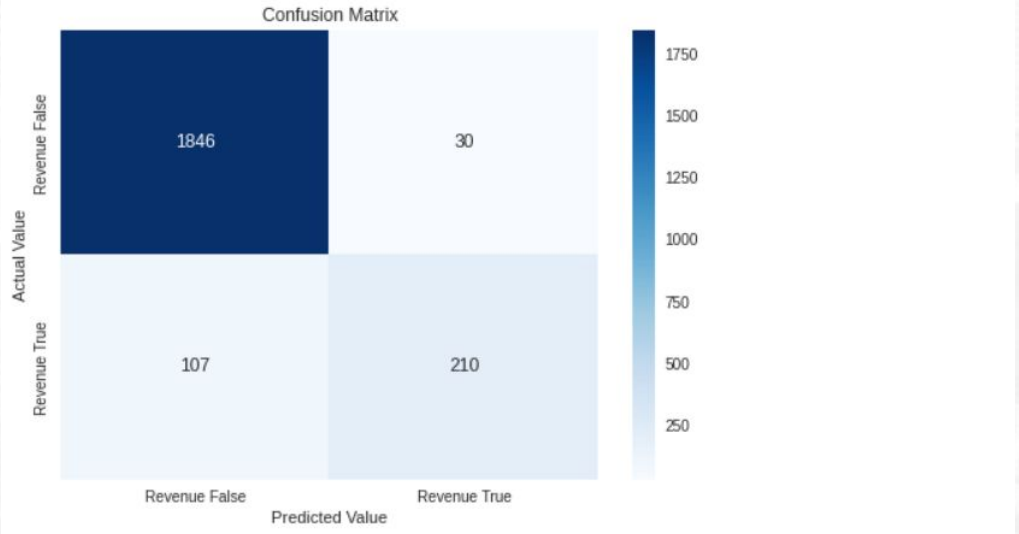
```
np.random.seed(28)
model = LGBMClassifier(random_state=28)

param_grid = {
    'boosting_type':['gbdt', 'dart', 'goss', 'rf'],
    'objective':['regression', 'binary', 'multiclass', 'lambdarank'],
    # 'max_depth':np.arange(5,28),
    # 'num_leaves':np.arange(5,20),
}

lgbmc = GridSearchCV(model, param_grid=param_grid, scoring='precision', cv=10)
lgbmc.fit(x_train, y_train)
```

1E. Hyperparameter Tuning (Hasil)

```
print('best params : ', lgbmc.best_params_)
predict_test_tuned = predict_model(lgbmc, raw_score=True)
plot_confusion_matrix(confusion_matrix(predict_test_tuned.Revenue, predict_test_tuned.Label))
```

```
best params :  {'boosting_type': 'dart', 'objective': 'regression'}
```

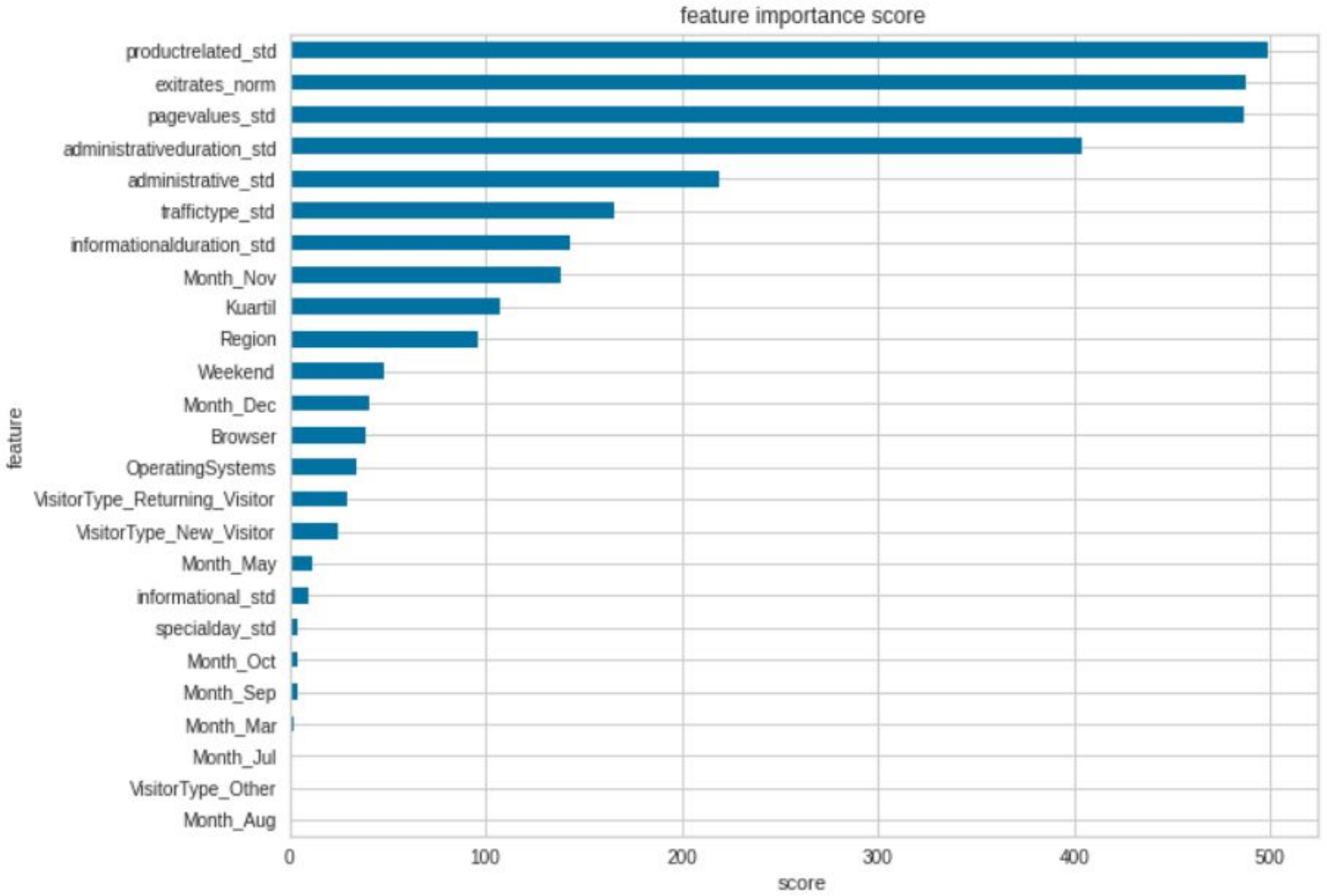
	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Light Gradient Boosting Machine	0.9375	0.9608	0.6625	0.875	0.754	0.719	0.7282



- Model hasil Hyperparameter Tuning memiliki nilai Accuracy Test, AUC Test, dan Precision Test lebih tinggi dari nilai Train-nya dengan perbedaan nilai yang kecil. Menandakan bahwa model hasil hypertuning **tidak overfitting maupun underfitting**.
- Model yang dipilih dapat **mendeteksi dengan baik** kepada pengungjung yang tidak menghasilkan Revenue yang dapat terlihat dari nilai **Precision yang tinggi**. Dari nilai **AUC**-nya, model **memiliki nilai yang tinggi** menandakan bahwa **performa model sudah bagus**.

Feature Importance

2A. Feature importance



Dari feature importance tersebut dapat dilihat bahwa feature ProductRelated memiliki pengaruh paling tertinggi. ProductRelated adalah jumlah halaman produk terkait yang dikunjungi oleh user. Dengan demikian untuk kedepannya kita dapat menampilkan rekomendasi product-product terkait yang dikunjungi user dengan lebih banyak untuk meningkatkan kemungkinan user untuk membeli.

2B. Feature selection dari Feature Important & Iterasi Model

Tim kami akan memilih 5 feature teratas yang dihasilkan oleh feature importance untuk dilakukan modeling kembali. Feature tersebut adalah ProductRelated, ExitRates, PageValues, AdministrativeDuration, dan Administrative. Modeling ulang ini kami lakukan dengan persentasi split yang sama, metode smote, dan menggunakan hyperparameter tuning kembali

```
best params :  {'boosting_type': 'dart', 'objective': 'regression'}
```

	data	accuracy	precision	roc_auc
0	train	0.944121	0.914550	0.822921
1	test	0.890105	0.690821	0.706353

Setelah dilakukan iterasi model dan dievaluasi kembali. Hasilnya menunjukan bahwa model iterasi ini mengalami **overfitting** yang dapat dilihat dari nilai accuracy, precision, dan roc_auc yang lebih besar pada data train dibangkan data test. Perbedaan nilainya pun cukup signifikan. Dengan demikian **model hasil iterasi tidak lebih baik dari model semula**

Rakamin Academy

Git

3. Repository Git

Berikut adalah Link Repository Git yang berisikan file-file final project dan readme.md

https://github.com/ROMBombo/tim_uranus_fp_rakamin

Pembagian Tugas (Stage 3)

1. Rijal Abdulhakim : Git, Modeling, Evaluation, Hyperparameter Tuning, Feature Importance, penyusunan laporan
2. Yusuf Rifqi H : Modelling, Feature Importance, Penyusun Notula
3. M Zamzam I : Modelling, Penyusunan laporan
4. Sahel Abdat : Modelling, EvaluAtion, Feature Importance
5. Putrini Nur A H : Penyusun Notula, Modelling, Evaluation, Feature importance
6. Surya Karunia R : Modelling, Evaluation
7. M Rendra Putra P : Modelling, Penyusunan Laporan