LAPORAN HOMEWORK SUPERVISED LEARNING

ANGGOTA KELOMPOK

Celestial Randy

Sonia Epifany Sandah

Oky Hariawan

Risca Naquitasia

Mochamad Choiril Iman

Ahmad Reza

Yehezkiel Novianto A.

MODELING

- Split Data Train & Test
- Model Evaluation
- Modeling
- Hyperparameter Tuning

SPLIT DATA TRAIN & TEST

What We Do:

- Melakukan split data antara Feature dan Target yang direpresentasikan dengan variable X dan y

```
[ ] # Split Feature and Label
    X = df_clean[['hotel','adults','is_repeated_guest','previous_cancellations','previous_bookings_not_canceled','reserved_room_type','assigned_room_type','booking_changes','days_in_waiting_list',
     'required_car_parking_spaces','total_of_special_requests','lead_time_norm','adr_norm',
     'distribution_channel_Corporate',
     'distribution_channel_Direct',
     'distribution_channel_GDS',
     'distribution_channel_TA/TO',
     'distribution_channel_Undefined',
     'deposit_type_No Deposit',
     'deposit_type_Non Refund',
     'deposit_type_Refundable',
     'customer_type_Contract',
     'customer_type_Group',
     'customer_type_Transient',
     'customer_type_Transient-Party','total_stays',
     'total_guests','kids','arrival_date_year',
     'arrival_date_week_number_norm',
     'arrival_date_day_of_month_norm','guest_location','meal','market_segment',
    ]]

    y = df_clean['is_canceled'] # target / label

    #Splitting the data into Train and Test
    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)
```

MODELING + MODEL EVALUATION (LIGHTGBM)

What We Do:

- Melakukan modeling dengan beberapa model salah satunya yaitu LightGBM

Insight:

- Model dengan default parameter ini menghasilkan score seperti gambar disamping
- Terlihat jika skor training model LightGBM sebesar 87% sangat dekat dengan skor testing 86,4%, yang berarti bahwa model tidak overfit atau underfit .

```
Accuracy (Test Set): 0.8641
Precision (Test Set): 0.8494
Recall (Test Set): 0.7716
F1-Score (Test Set): 0.8086
AUC: 0.9417
```

```
Training set score: 0.8709
Test set score: 0.8641
```

MODELING + MODEL EVALUATION (LOGISTIC REGRESSION)

What We Do:

- Melakukan modeling dengan beberapa model salah satunya yaitu Logistic Regression

Insight:

- Model dengan default parameter ini menghasilkan score seperti gambar disamping
- Terlihat jika skor training model logistic regression sebesar 80.6% sangat dekat dengan skor testing 80.3%, yang berarti bahwa model tidak overfit atau underfit

```
Accuracy (Test Set): 0.80
Precision (Test Set): 0.81
Recall (Test Set): 0.62
F1-Score (Test Set): 0.70
AUC: 0.88
```

```
Train score: 0.8062223285868134
Test score:0.8031605986151441
```

MODELING + MODEL EVALUATION (RESULT)

What We Do:

- Mencari model dengan nilai *Precision* tertinggi dari semua model yang ada

Algorithm	Precision (Test)	Accuracy (Test Set)	Recall (Test Set)	F1-Score
Logistic Regression	81%	80%	62%	70%
XGBoost	83%	84%	71%	77%
KNN	79%	84%	76%	78%
Adaboost	81%	82%	68%	74%
LightGBM	86%	88%	81%	83%

Insight:

- Model terbaik dari hasil evaluasi yang dipilih adalah LightGBM yang memberikan performa terbaik.

HYPERPARAMETER TUNING (1)

What We Do:

- Melakukan Hyperparameter Tuning dengan bantuan *framework* Optuna
- mencoba hyperparameter tuning dengan parameter objective, metric, num_boost_round, dan learning rate. nanti optuna akan mencoba coba kombinasi parameter yang tepat dan melakukan trial sejumlah yang kita masukkan (disini n_trial yang digunakan 100).

```
import optuna

def objective_lgbm(trial):

    param = {
        'objective': 'binary',
        'metric': 'accuracy_score',
        'num_leaves': trial.suggest_int('num_leaves', 10,100),
        'num_boost_rounds': trial.suggest_int('num_boost_rounds', 100,300),
        'learning_rate': trial.suggest_loguniform('learning_rate', 0.1,1)
    }

    LightGBM_Manual = lgb.train(param, train_data, num_boost_rounds)
    preds=LightGBM_Manual.predict(X_test)
    pred_labels = np.rint(preds)
    accuracy = round(accuracy_score(y_test, pred_labels),4)
    return accuracy

study_lgbm = optuna.create_study(direction='maximize',study_name="LGBM")
study_lgbm.optimize(objective_lgbm, n_trials=100)
```

```
Model Accuracy -->  0.883
Model's Best parameters -->  {'num_leaves': 96, 'num_boost_rounds': 107, 'learning_rate': 0.26587222534561905}
```

Insight:

- Mendapat parameter apa saja yang dapat digunakan untuk mendapatkan hasil akurasi terbaik.

HYPERPARAMETER TUNING (2)

What We Do:

- Membuat model dengan parameter terbaik dan melihat *score* modelnya

Insight:

Dapat dilihat setelah melakukan hypertuning score presisi meningkat menjadi 85.8% dari nilai sebelumnya 84%. Terlihat pula jika skor training model LightGBM setelah hyperparameter tuning sebesar 92.9% dekat dengan skor testing 87,9% yang berarti bahwa model tidak overfit atau underfit.

```
[ ]  # build the lightgbm model
     model_LGBM = lgb.LGBMClassifier(**trial_lgbm.params)
     fit_model_LGBM = model_LGBM.fit(X_train,y_train)
     pred_LGBM = fit_model_LGBM.predict(X_test)
     accuracy=accuracy_score(y_test, pred_LGBM)
     print('LightGBM Model accuracy score: {0:0.4f}'.format(accuracy_score(y_test, pred_LGBM)))

     LightGBM Model accuracy score: 0.8796
```

```
[ ]  y_pred_train = fit_model_LGBM.predict(X_train)
     print('Training-set accuracy score: {0:0.4f}'. format(accuracy_score(y_train, y_pred_train)))

     Training-set accuracy score: 0.9295
```

```
Accuracy (Test Set): 0.8796
Precision (Test Set): 0.8586
Recall (Test Set): 0.8097
F1-Score (Test Set): 0.8334
```

FEATURE IMPORTANCE

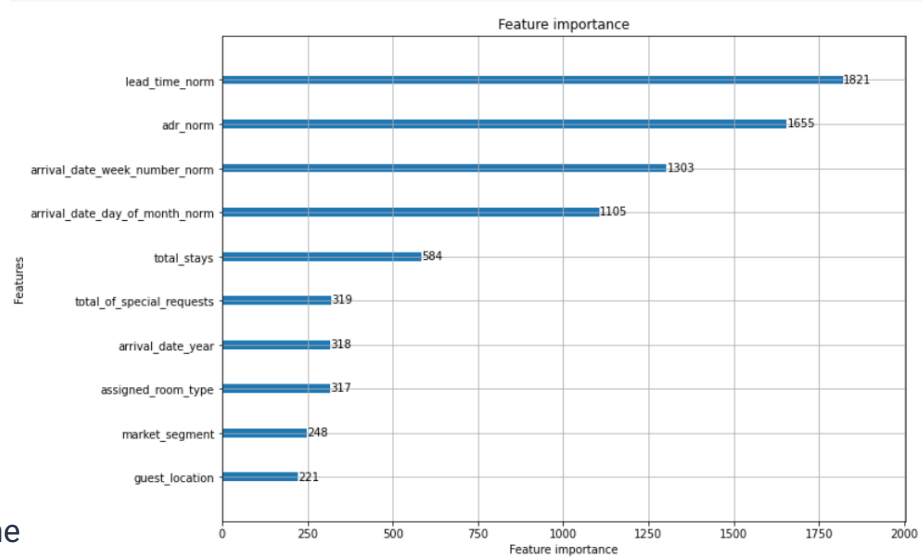
- Feature Importance
- Feature Selection

FEATURE IMPORTANCE (MENCARI TOP 10 FEATURE)

Insight:

10 Top Feature:

- 1. Lead Time
- 2. Adr
- 3. Arrival date week number
- 4. Arrival date day of month
- 5. Total stays
- 6. Total of Special Request
- 7. Arrival Date Year
- 8. Assigned Room Type
- 9. Market Segment
- 10. Guest Location



Business Recommendation:

Dari hasil feature importance dapat terlihat bahwa Lead time mempunyai pengaruh terbesar dalam pembatalan pesanan hotel. Maka dari itu pihak hotel dapat menerapkan batasan maksimal waktu pemesanan kamar dan menerapkan deposit/uang muka untuk reservasi dengan jangka waktu lama guna menurunkan tingkat pembatalan pesanan.

FEATURE SELECTION
(HASIL MODELING DENGAN TOP FEATURE)

Insight:

Dapat dilihat setelah melakukan feature selection score presisi sebesar 83%. Terlihat pula jika skor training model LightGBM dengan feature selection sebesar 90.2% dekat dengan skor testing 85,4% yang berarti bahwa model tidak overfit atau underfit.

```
# accuracy
# check for overfitting
# print the scores on training and test set
print('Training set score: {:.4f}'.format(model_LGBM.score(X_train_new, y_train)))
print('Test set score: {:.4f}'.format(model_LGBM.score(X_test_new, y_test)))
```

```
Training set score: 0.9021
Test set score: 0.8541
```

```
# precision recall f1-score
print("Precision (Test Set): %.4f" % precision_score(y_test, pred_LGBM))
print("Recall (Test Set): %.4f" % recall_score(y_test, pred_LGBM))
print("F1-Score (Test Set): %.4f" % f1_score(y_test, pred_LGBM))
```

```
Precision (Test Set): 0.8312
Recall (Test Set): 0.7628
F1-Score (Test Set): 0.7955
```

GITHUB COLLABORATION

https://github.com/celestialrandy/rakamin-project

TERIMA KASIH