UTS

MACHINE LEARNING

Oleh :

M. Rakan Bagus / 1103213162

PRODI S1 TEKNIK KOMPUTER

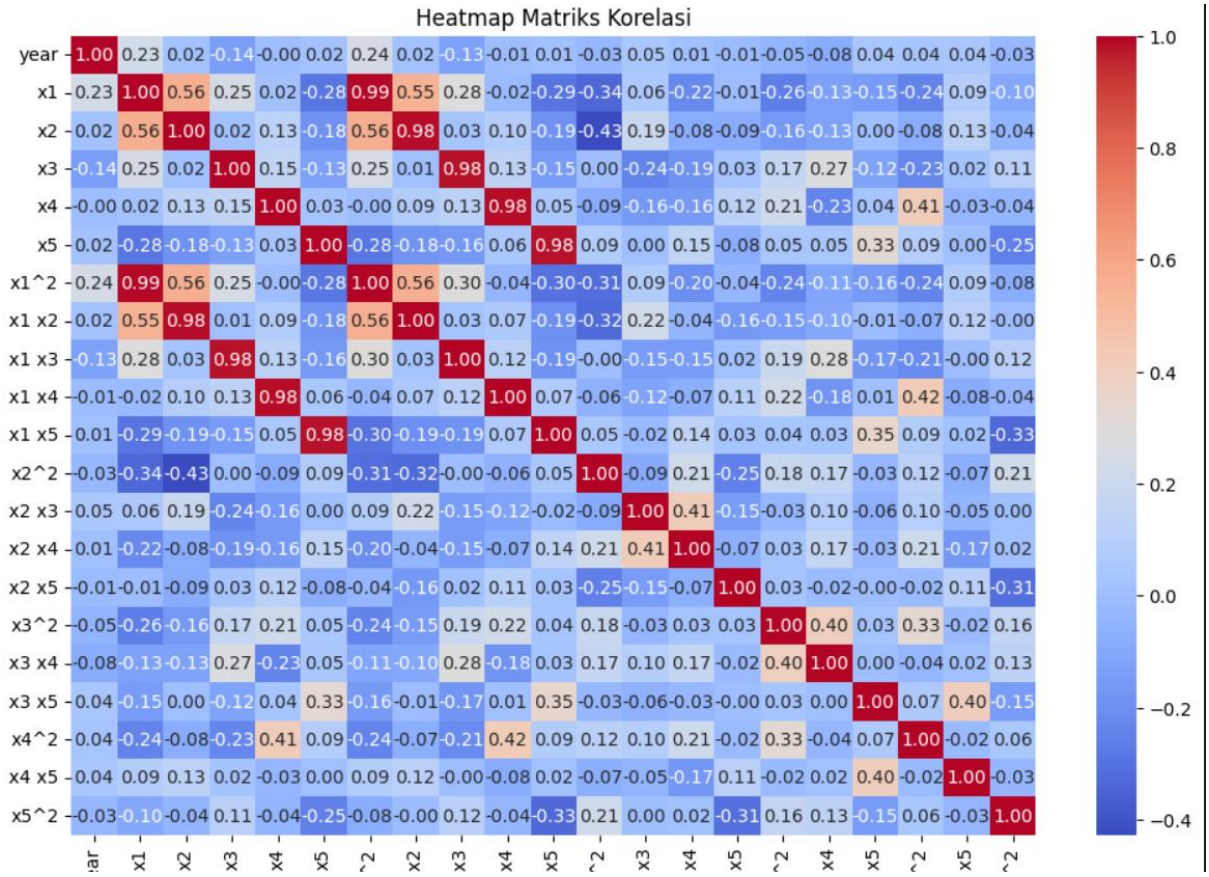
FAKULTAS TEKNIK ELEKTRO

UNIVERSITAS TELKOM

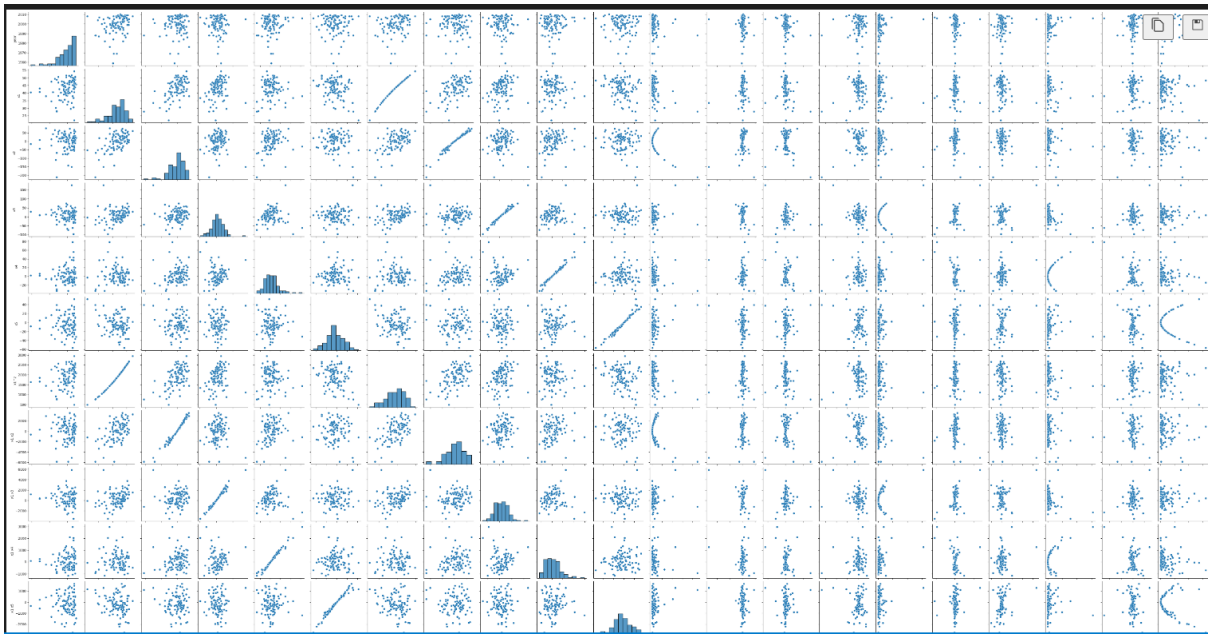
BANDUNG

2024

1. **Regression Model (Dataset RegresiUTSTelkom.csv)**
    - **Exploratory Data Analysis (EDA) & Data Visualization**



Heatmap adalah Fitur dengan korelasi tinggi terhadap target lebih relevan untuk model.



Scatter Plot untuk menunjukkan apakah data target terdistribusi normal atau bias ke nilai tertentu.

- **Pipeline dengan sckit-learn**

```
# Membuat laporan performa
results = {
    "Model": ["Polynomial Regression", "Decision Tree", "k-NN Regression", "XGBoost Regression"],
    "MSE": [mse_poly, mse_tree, mse_knn, mse_xgb],
    "R2 Score": [r2_poly, r2_tree, r2_knn, r2_xgb]
}

results_df = pd.DataFrame(results)
print(results_df)
```

Output :

```
                Model         MSE  R2 Score
0  Polynomial Regression  202.461972 -0.701146
1          Decision Tree  199.597202 -0.677076
2        k-NN Regression  113.474899  0.046550
3     XGBoost Regression   96.682911  0.187641
```

Kesimpulan Berdasarkan Hasil Evaluasi Model

Berikut adalah analisis hasil evaluasi untuk keempat model berdasarkan nilai Mean Squared Error (MSE) dan R2 Score:

1. Polynomial Regression

MSE: 202.46 (tertinggi di antara semua model).

R2 Score: -0.70 (nilai negatif menunjukkan model tidak mampu menjelaskan variansi data dengan baik).

Analisis:

- Polynomial Regression tidak cocok untuk dataset ini. Kombinasi fitur polynomial yang terlalu banyak mungkin menyebabkan overfitting pada data latih tetapi gagal generalisasi pada data uji.
- Kemungkinan derajat polynomial terlalu tinggi, atau data memerlukan preprocessing tambahan untuk meningkatkan kinerjanya.

2. Decision Tree

MSE: 199.59 (hampir sama dengan Polynomial Regression, tetapi sedikit lebih baik).

R2 Score: -0.68 (masih negatif, menunjukkan performa buruk pada data uji).

Analisis:

- Decision Tree cenderung overfit pada dataset ini.
- Meskipun lebih baik dibandingkan Polynomial Regression, model ini juga tidak mampu menangkap pola yang relevan secara efektif.
- Parameter seperti kedalaman pohon (max_depth) atau minimal jumlah sampel perlu diatur melalui hyperparameter tuning.

3. k-NN Regression
MSE: 113.47 (penurunan signifikan dibandingkan Polynomial dan Decision Tree).
R2 Score: 0.05 (positif, menunjukkan model mulai dapat menjelaskan variansi data meskipun masih sangat lemah).
Analisis:

- k-NN menunjukkan performa yang lebih baik dibandingkan dua model sebelumnya.
- Karena sensitif terhadap jarak antar data, performanya meningkat setelah data distandarisasi.
- Model ini cocok untuk hubungan lokal antar fitur, tetapi mungkin memerlukan optimasi pada jumlah tetangga (n_neighbors) untuk hasil yang lebih baik.

4. XGBoost Regression
MSE: 96.68 (terendah di antara semua model, menunjukkan kesalahan prediksi yang paling kecil).
R2 Score: 0.19 (tertinggi, menunjukkan model mampu menjelaskan sekitar 19% variansi data).
Analisis:

- XGBoost adalah model dengan performa terbaik pada dataset ini. Kombinasi teknik ensemble dan kemampuannya menangani hubungan non-linear membuatnya unggul.
- Dengan hyperparameter tuning (misalnya, learning_rate, max_depth, dan n_estimators), performa XGBoost dapat lebih ditingkatkan.

- **HYPERPARAMETER TUNING PADA SETIAP MODEL**

```
# Membuat laporan hasil hyperparameter tuning dan evaluasi model
results = {
    "Model": ["Polynomial Regression", "Decision Tree", "k-NN Regression", "XGBoost Regression"],
    "Best Parameters": [
        grid_poly.best_params_,
        grid_tree.best_params_,
        grid_knn.best_params_,
        grid_xgb.best_params_,
    ],
    "MSE": [mse_poly, mse_tree, mse_knn, mse_xgb],
    "R2 Score": [r2_poly, r2_tree, r2_knn, r2_xgb],
}

# Membuat DataFrame hasil
results_df = pd.DataFrame(results)
print(results_df)
```

```
                 Model                                  Best Parameters  \
0  Polynomial Regression        {'model__max_depth': 10, 'poly__degree': 2}
1          Decision Tree  {'model__max_depth': 10, 'model__min_samples_s...
2        k-NN Regression  {'model__n_neighbors': 5, 'model__weights': 'u...
3     XGBoost Regression  {'model__learning_rate': 0.1, 'model__max_dept...

          MSE  R2 Score
0  100.118780  0.158772
1   99.573058  0.163357
2  113.474899  0.046550
3   96.230511  0.191442
```

Kesimpulan dari Hyperparameter Tuning :

1. Polynomial Regression

Best Parameters: {'model__max_depth': 10, 'poly__degree': 2}

MSE: 100.12 (cukup baik, namun tidak terbaik).

R2 Score: 0.16 (masih rendah, model hanya menjelaskan 16% dari variansi data).

2. Decision Tree

Best Parameters: {'model__max_depth': 10, 'model__min_samples_split': 5}

MSE: 99.57 (lebih baik dari Polynomial Regression).

R2 Score: 0.16 (sebanding dengan Polynomial Regression).

3. k-NN Regression

Best Parameters: {'model__n_neighbors': 5, 'model__weights': 'uniform'}

MSE: 113.47 (tertinggi di antara semua model, menunjukkan kesalahan prediksi yang besar).

R2 Score: 0.05 (paling rendah, hanya menjelaskan 5% dari variansi data).

4. XGBoost Regression

Best Parameters: {'model__learning_rate': 0.1, 'model__max_depth': 5, 'model__n_estimators': 100}

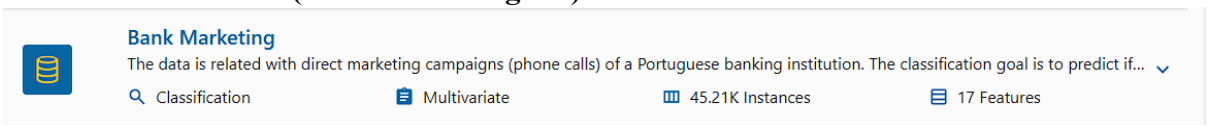
MSE: 96.23 (terendah di antara semua model, menunjukkan performa prediksi terbaik).

R2 Score: 0.19 (tertinggi, menjelaskan 19% dari variansi data).

KESIMPULAN AKHIR

Gunakan XGBoost sebagai Model Akhir: Dengan performa terbaik, XGBoost menunjukkan hasil paling konsisten. XGBoost Terbaik di antara semua model

2. Classification Model (BankMarketing.csv)



- **Explanatory Data Analysis (EDA)**

```
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 17 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   age        45211 non-null  int64
 1   job        45211 non-null  object
 2   marital    45211 non-null  object
 3   education  45211 non-null  object
 4   default    45211 non-null  object
 5   balance    45211 non-null  int64
 6   housing    45211 non-null  object
 7   loan       45211 non-null  object
 8   contact    45211 non-null  object
 9   day        45211 non-null  int64
 10  month      45211 non-null  object
 11  duration   45211 non-null  int64
 12  campaign   45211 non-null  int64
 13  pdays      45211 non-null  int64
 14  previous   45211 non-null  int64
 15  poutcome   45211 non-null  object
 16  y          45211 non-null  object
dtypes: int64(7), object(10)
memory usage: 5.9+ MB
None
...
1  unknown    5   may       151         1      -1         0  unknown  no
2  unknown    5   may        76         1      -1         0  unknown  no
3  unknown    5   may        92         1      -1         0  unknown  no
4  unknown    5   may       198         1      -1         0  unknown  no
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

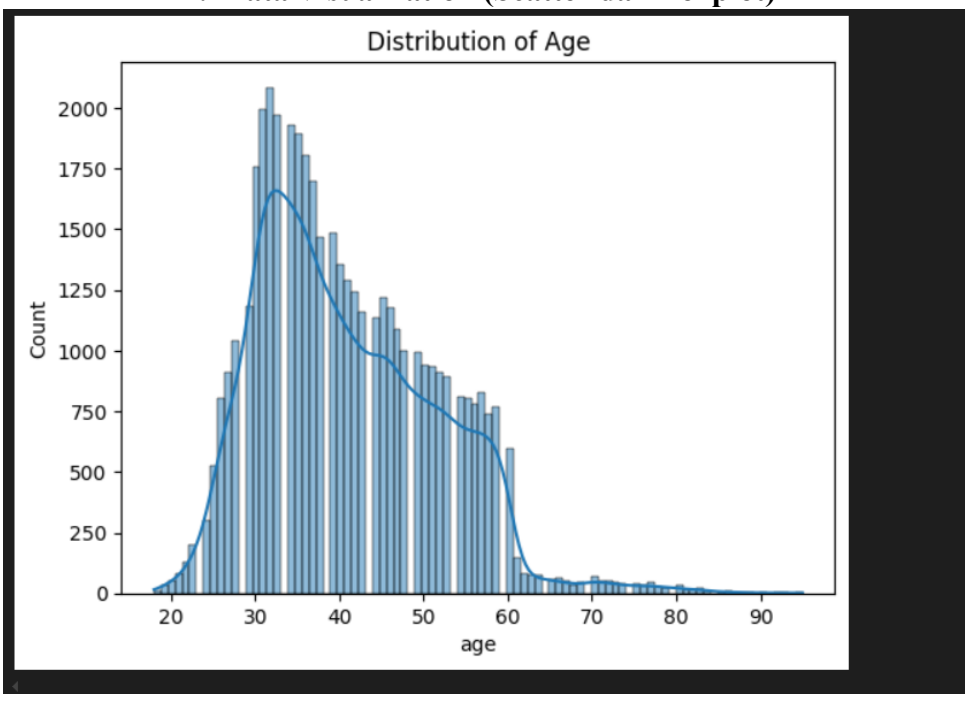
Dataset terdiri dari 45.211 entri dengan 17 kolom, mencakup data numerik dan kategorikal.

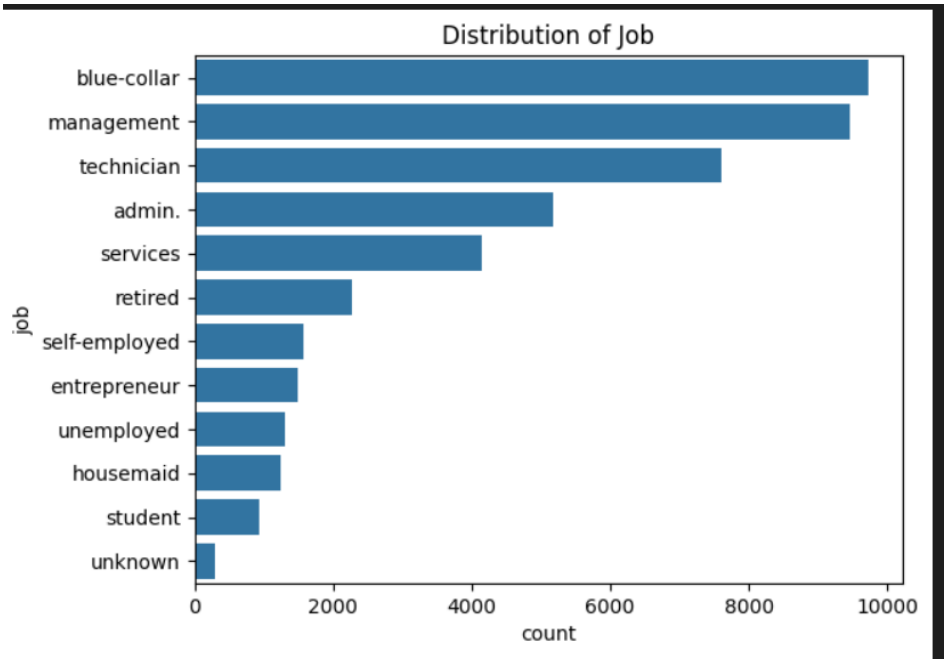
```python
# Check Missing Values
# Memeriksa apakah ada nilai yang hilang di dataset
print(data.isnull().sum())
```

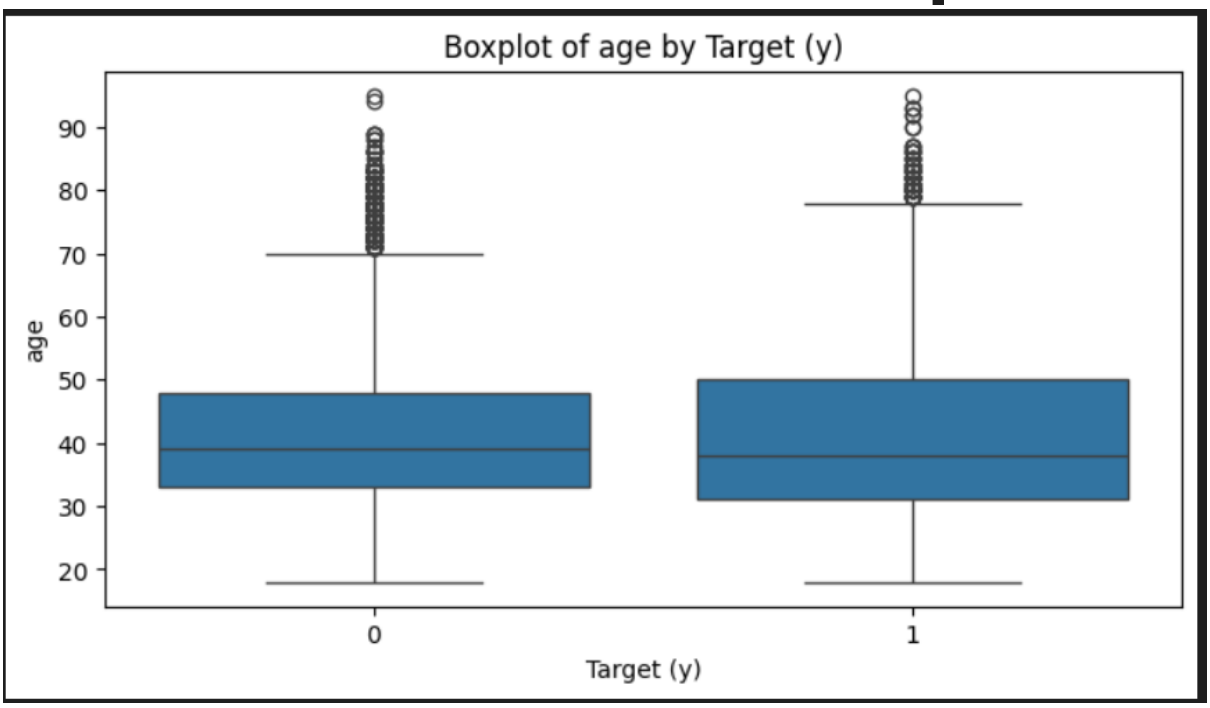
```
age          0
job          0
marital      0
education    0
default      0
balance      0
housing      0
loan         0
contact      0
day          0
month        0
duration     0
campaign     0
pdays        0
previous     0
poutcome     0
y            0
dtype: int64
```

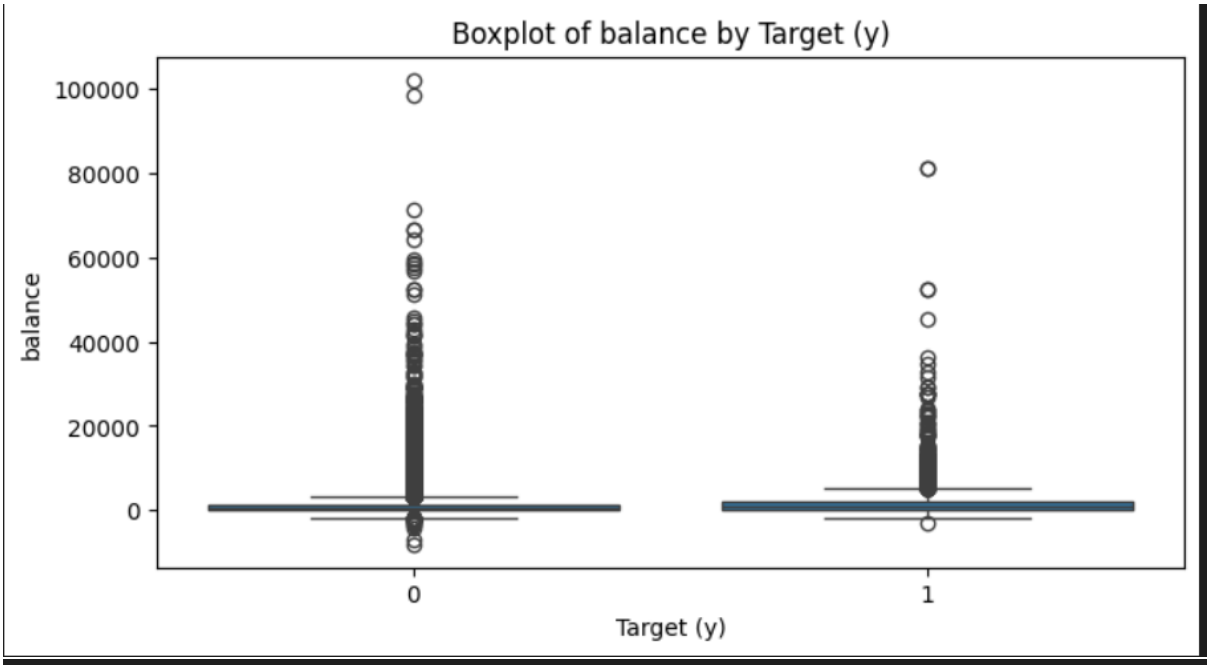
Tidak ada nilai yang hilang di dataset, sehingga tidak perlu penanganan tambahan untuk missing values.

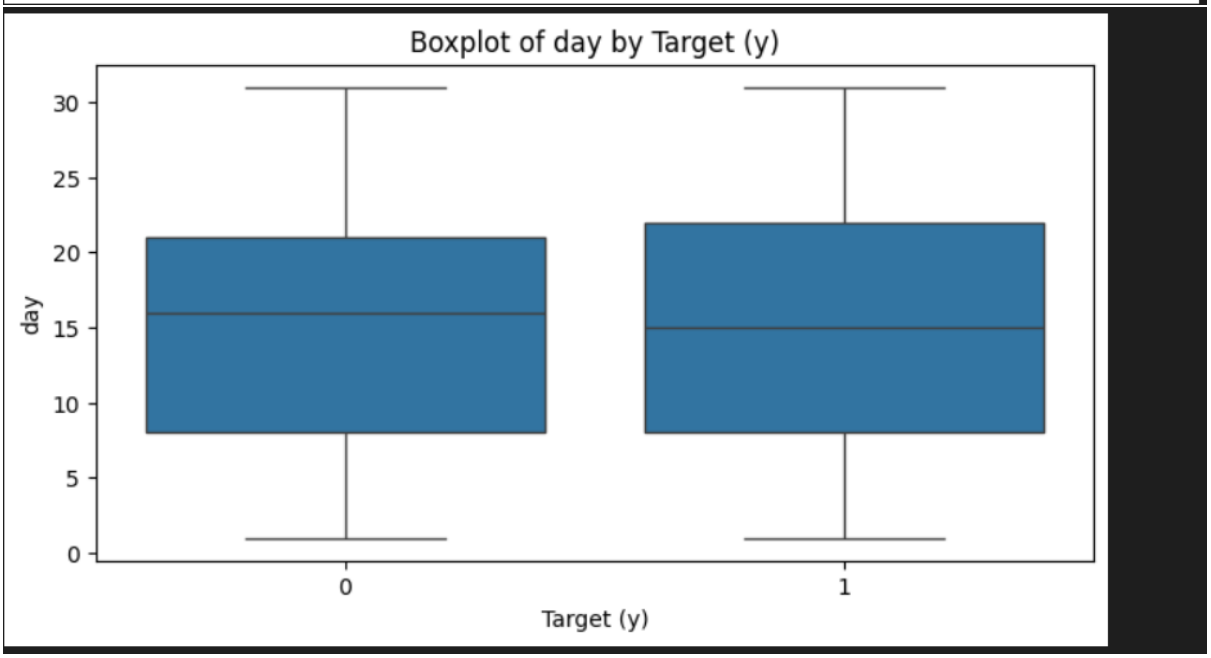
- **2. Data Visualization (Scatter dan Boxplot)**

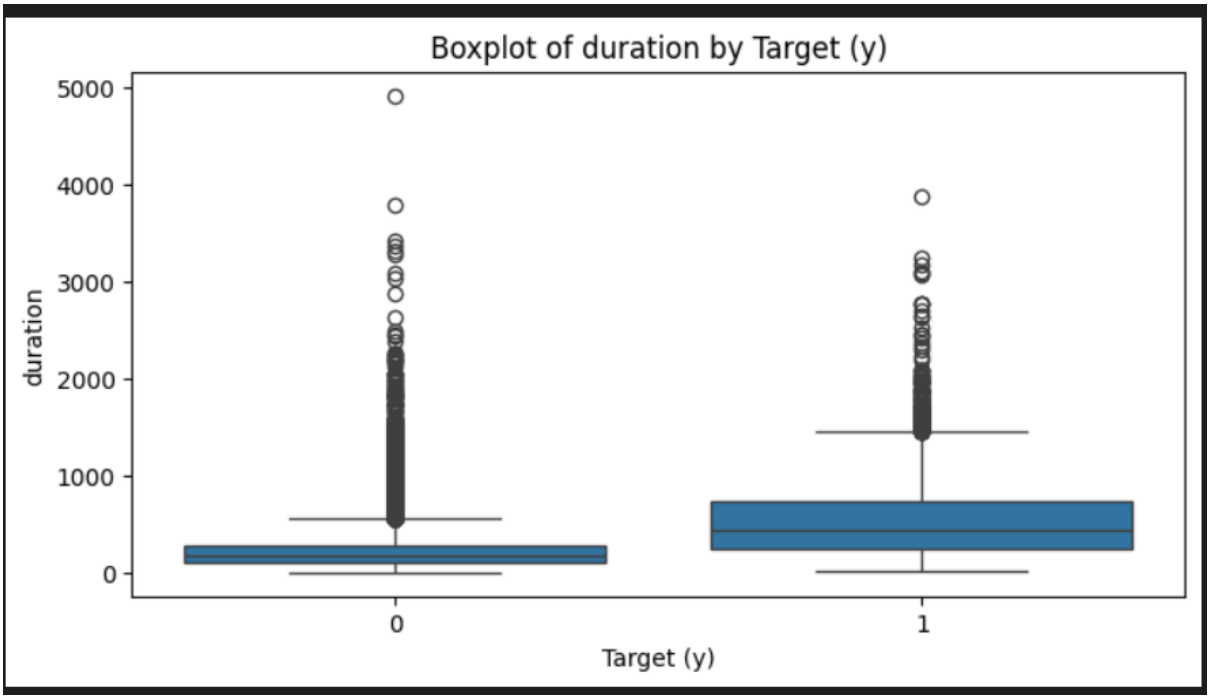


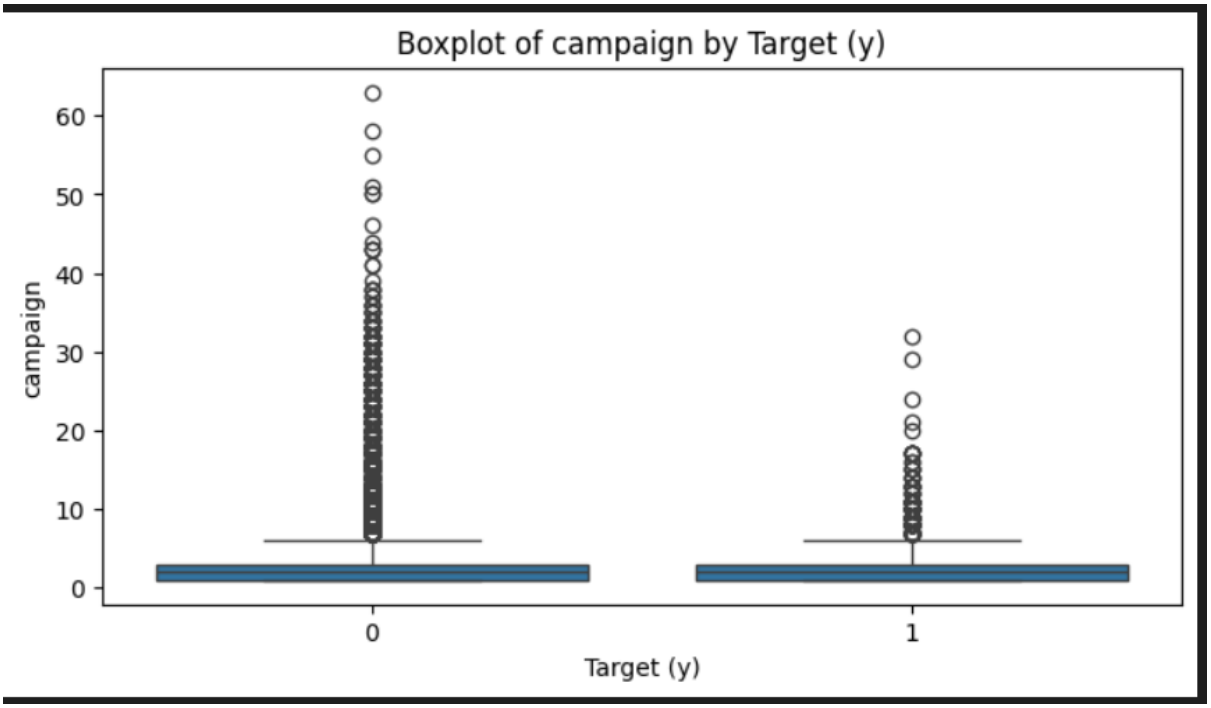

Distribution of Job
blue-collar
management
technician
admin.
services
retired
self-employed
entrepreneur
unemployed
housemaid
student
unknown
job
0
2000
4000
6000
8000
10000
count

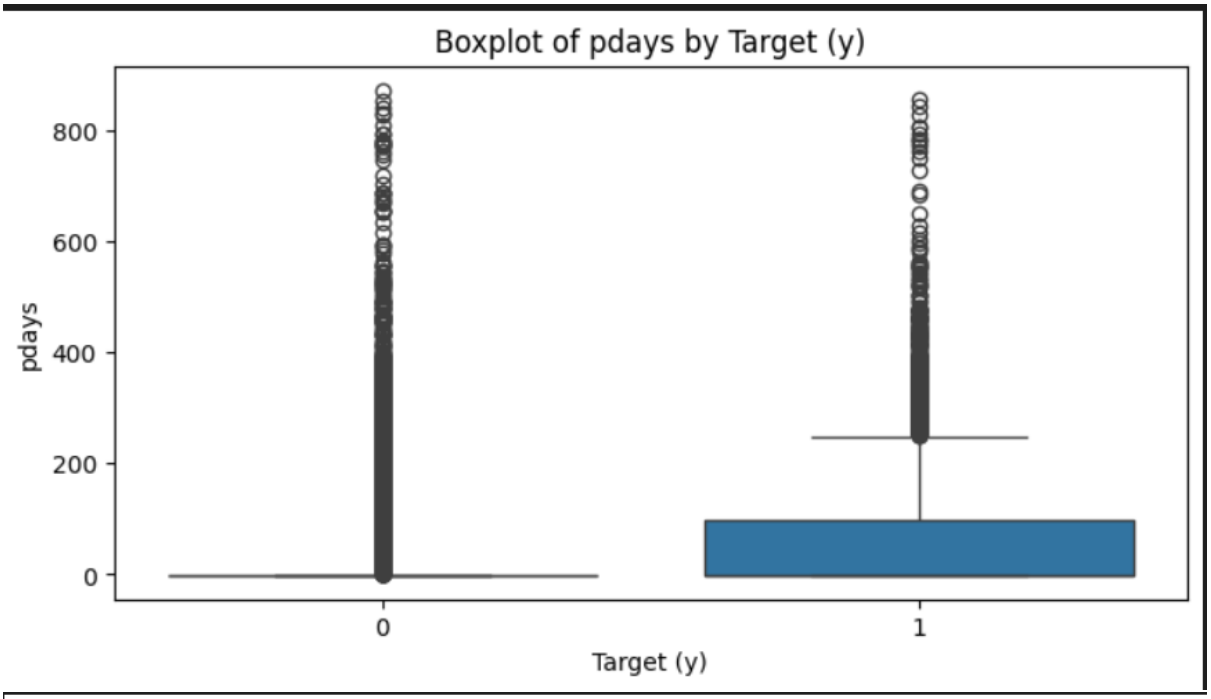

Boxplot of age by Target (y)
90
80
70
60
50
40
30
20
age
0
1
Target (y)

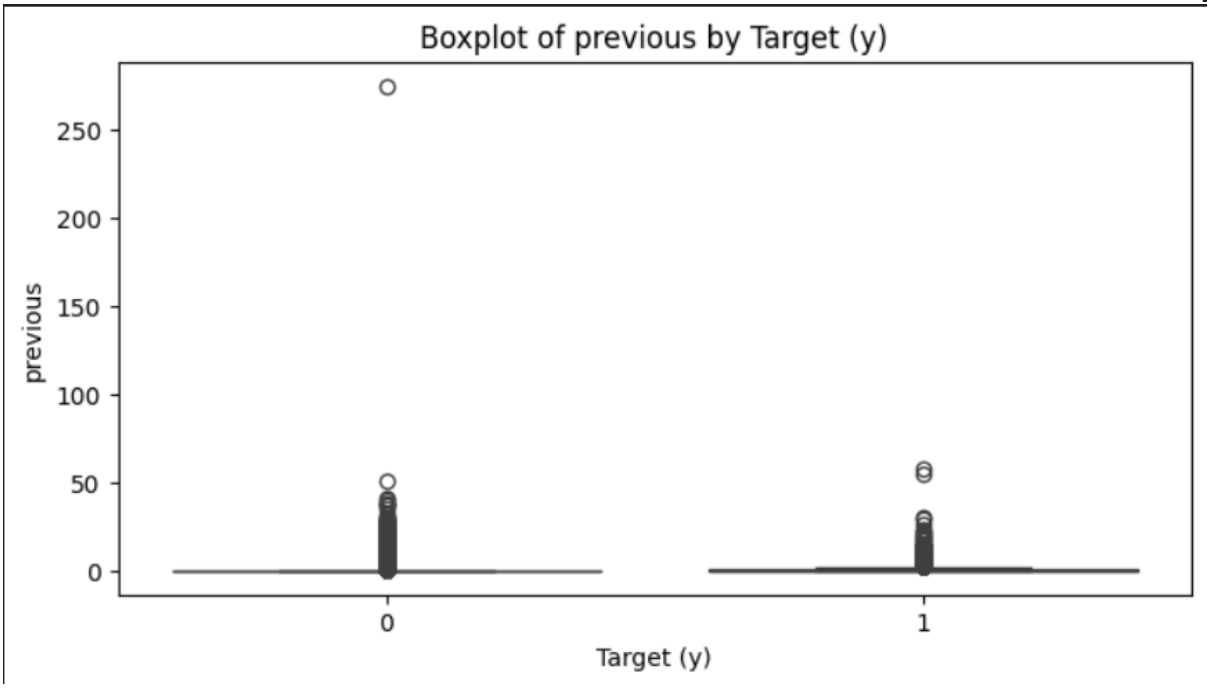

Boxplot of balance by Target (y)
balance
100000
80000
60000
40000
20000
0
0
1
Target (y)


Boxplot of day by Target (y)
day
30
25
20
15
10
5
0
0
1
Target (y)


Boxplot of duration by Target (y)
duration
5000
4000
3000
2000
1000
0
0
1
Target (y)


Boxplot of campaign by Target (y)
campaign
60
50
40
30
20
10
0
0
1
Target (y)





- Pada grafik Distribution of age :

Distribusi umur memiliki puncak di sekitar usia 30-40 tahun, menunjukkan mayoritas nasabah berada di usia produktif.

- Pada grafik Job Distribution :

Pekerjaan yang paling umum adalah "blue-collar", diikuti oleh "management" dan "technician".

- Pada Boxplot :

1. Nasabah dengan durasi lebih tinggi cenderung memiliki target 'yes' (y=1).
2. Tidak ada pola signifikan pada variabel lain seperti 'campaign' atau 'pdays'.

- **Pipeline dengan scikit-learn**

```
# Train and Evaluate Models
# Melatih dan mengevaluasi setiap model
results = {}
for name, model in models.items():
    clf = Pipeline(steps=[('preprocessor', preprocessor),
                          ('classifier', model)])
    clf.fit(X_train, y_train)  # Melatih model
    y_pred = clf.predict(X_test)  # Memprediksi data uji
    acc = accuracy_score(y_test, y_pred)  # Menghitung akurasi
    results[name] = acc
    print(f"{name} Accuracy: {acc:.2f}")
    print(classification_report(y_test, y_pred))
```

```
Logistic Regression Accuracy: 0.90
              precision    recall  f1-score   support

           0       0.92      0.97      0.95      7985
           1       0.64      0.35      0.45      1058

    accuracy                           0.90      9043
   macro avg       0.78      0.66      0.70      9043
weighted avg       0.89      0.90      0.89      9043

Decision Tree Accuracy: 0.87
              precision    recall  f1-score   support

           0       0.93      0.93      0.93      7985
           1       0.46      0.48      0.47      1058

    accuracy                           0.87      9043
   macro avg       0.70      0.70      0.70      9043
weighted avg       0.88      0.87      0.87      9043

k-NN Accuracy: 0.90
              precision    recall  f1-score   support

           0       0.92      0.97      0.94      7985
           1       0.60      0.34      0.43      1058
...
    accuracy                           0.91      9043
   macro avg       0.78      0.73      0.75      9043
weighted avg       0.90      0.91      0.90      9043
```

Kesimpulan : Logistic Regression dan XGBoost cenderung memberikan akurasi lebih tinggi dibanding Decision Tree dan k-NN.

- **Hyperparameter Tuning dengan GridSearchCV**

```
# Hyperparameter Tuning
# Mendefinisikan parameter grid untuk setiap model
param_grid_lr = {'classifier__C': [0.1, 1, 10]}
param_grid_dt = {'classifier__max_depth': [3, 5, 10]}
param_grid_knn = {'classifier__n_neighbors': [3, 5, 7]}
param_grid_xgb = {'classifier__max_depth': [3, 5], 'classifier__learning_rate': [0.1, 0.01]}

# Melakukan tuning hyperparameter untuk setiap model
param_grids = [param_grid_lr, param_grid_dt, param_grid_knn, param_grid_xgb]

for name, model, param_grid in zip(models.keys(), models.values(), param_grids):
    grid_search = GridSearchCV(estimator=Pipeline(steps=[
        ('preprocessor', preprocessor),
        ('classifier', model)]), param_grid=param_grid, cv=5, scoring='accuracy')
    grid_search.fit(X_train, y_train)
    print(f"Best params for {name}: {grid_search.best_params_}")
    print(f"Best accuracy for {name}: {grid_search.best_score_:.2f}")
```

```
Best params for Logistic Regression: {'classifier__C': 10}
Best accuracy for Logistic Regression: 0.90
Best params for Decision Tree: {'classifier__max_depth': 5}
Best accuracy for Decision Tree: 0.90
Best params for k-NN: {'classifier__n_neighbors': 7}
Best accuracy for k-NN: 0.90
Best params for XGBoost: {'classifier__learning_rate': 0.1, 'classifier__max_depth': 5}
Best accuracy for XGBoost: 0.91
```

Tuning hyperparameter membantu meningkatkan akurasi dengan mencari kombinasi parameter terbaik.

Kesimpulan akhir dari Hyperparameter Tuning :

1. Logistic Regression

Parameter terbaik: C = 10

Akurasi terbaik: 90%

Analisis: Logistic Regression menunjukkan kinerja yang sangat baik dengan akurasi 90%. Model ini sederhana, cepat, dan cocok untuk data dengan hubungan linier antara fitur dan target.

2. Decision Tree

Parameter terbaik: max_depth = 5

Akurasi terbaik: 90%

Analisis: Decision Tree memberikan akurasi setara dengan Logistic Regression. Model ini lebih fleksibel dibanding Logistic Regression, namun sedikit lebih rentan terhadap overfitting.

3. k-NN (k-Nearest Neighbors)

Parameter terbaik: n_neighbors = 7

Akurasi terbaik: 90%

Analisis: k-NN memiliki akurasi yang sama dengan Logistic Regression dan Decision Tree, namun memerlukan waktu lebih lama untuk prediksi karena sifatnya sebagai lazy learner. Model ini lebih cocok untuk data yang sudah dinormalisasi dengan baik.

4. XGBoost
Parameter terbaik: learning_rate = 0.1, max_depth = 5
Akurasi terbaik: 91%
Analisis: XGBoost memberikan akurasi tertinggi di antara semua model. Model ini sangat kuat untuk menangani dataset kompleks dan besar. Namun, model ini lebih membutuhkan waktu dan sumber daya komputasi.

Kesimpulan dan Rekomendasi
XGBoost adalah model dengan performa terbaik (akurasi 91%) dan direkomendasikan jika prioritas adalah kinerja optimal, terutama pada dataset kompleks.
Jika interpretasi model atau efisiensi waktu lebih penting, Logistic Regression adalah pilihan yang baik karena memiliki akurasi tinggi (90%) dan lebih mudah dipahami serta diterapkan.