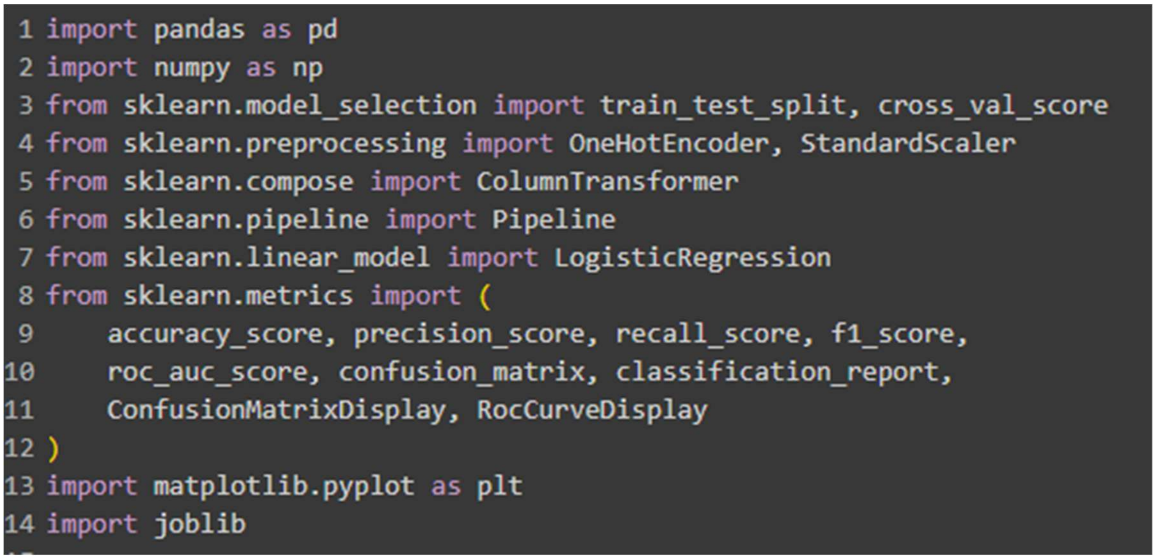
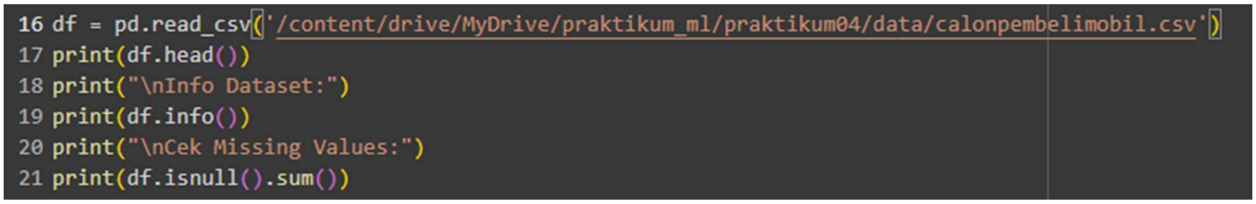
PRAKTIKUM 04

1. Import library yang akan digunakan



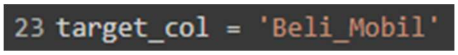
```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import (
    accuracy_score, precision_score, recall_score, f1_score,
    roc_auc_score, confusion_matrix, classification_report,
    ConfusionMatrixDisplay, RocCurveDisplay
)
import matplotlib.pyplot as plt
import joblib
```

2. Membaca dataset



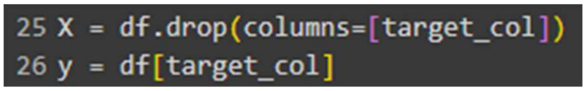
```python
df = pd.read_csv('/content/drive/MyDrive/praktikum_ml/praktikum04/data/calonpembelimobil.csv')
print(df.head())
print("\nInfo Dataset:")
print(df.info())
print("\nCek Missing Values:")
print(df.isnull().sum())
```

3. Menentukan kolom target
   Nilai 1 – calon pembeli membeli mobil
   Nilai 0 – calon pembeli tidak membeli mobil



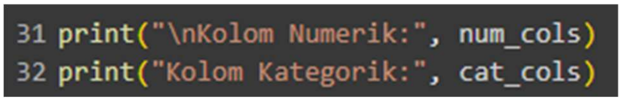
```python
target_col = 'Beli_Mobil'
```

4. Pisahkan fitur (X) dan target (Y)



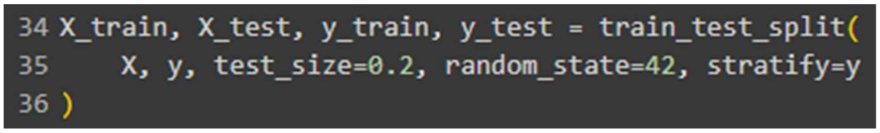
```python
X = df.drop(columns=[target_col])
y = df[target_col]
```

5. Tentukan kolom numerik dan ketegorikal



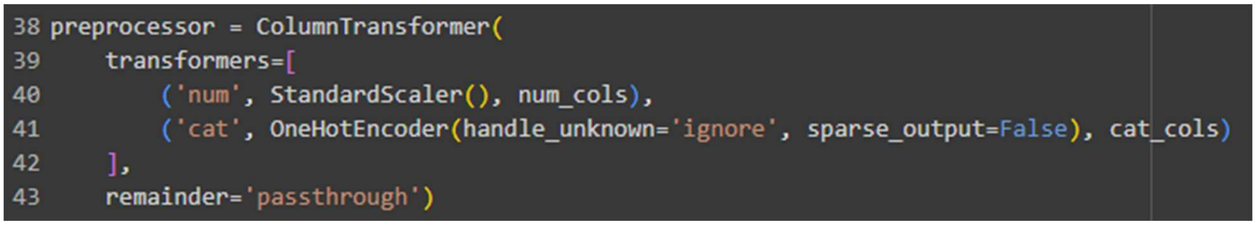
```python
print("\nKolom Numerik:", num_cols)
print("Kolom Kategorik:", cat_cols)
```

6. Membagi data menjadi train dan test



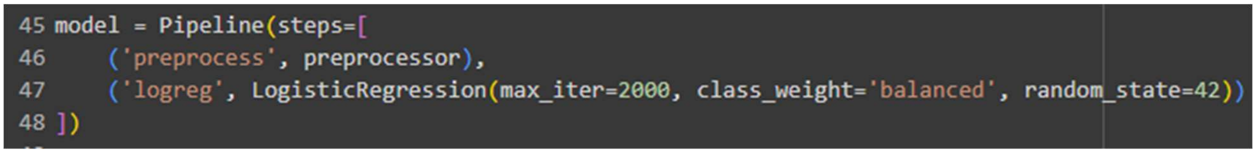
```python
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)
```

7. Preprocessing
   Standarisasi kolom mean = 0, std = 1
   Encoding mengubah kolom kategorikal menjadi format numerik biner



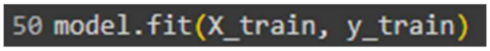
```
preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), num_cols),
        ('cat', OneHotEncoder(handle_unknown='ignore', sparse_output=False), cat_cols)
    ],
    remainder='passthrough')
```

8. Membuat pipe line model



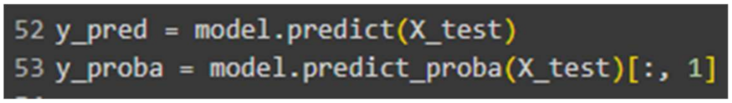
```
model = Pipeline(steps=[
    ('preprocess', preprocessor),
    ('logreg', LogisticRegression(max_iter=2000, class_weight='balanced', random_state=42))
])
```

9. Melatih model



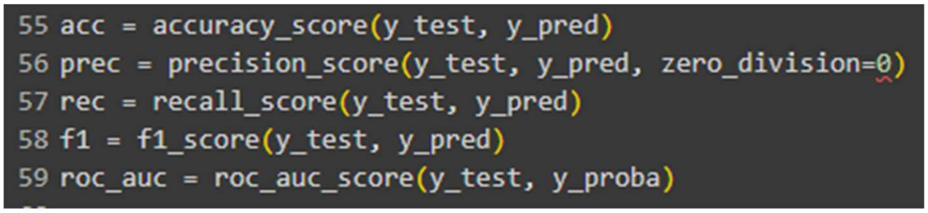
```
model.fit(X_train, y_train)
```

10. Melakukan prediksi



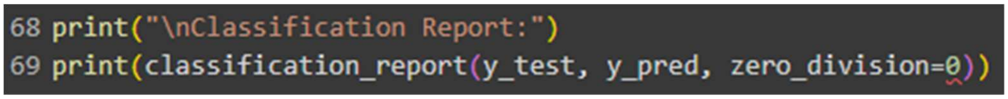
```
y_pred = model.predict(X_test)
y_proba = model.predict_proba(X_test)[:, 1]
```

11. Evaluasi model



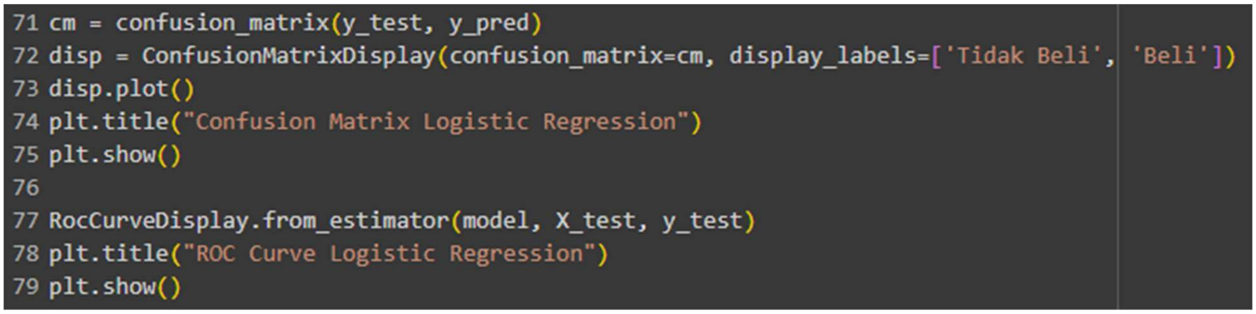
```
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred, zero_division=0)
rec = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
roc_auc = roc_auc_score(y_test, y_proba)
```

12. Menampilkan hasil evaluasi



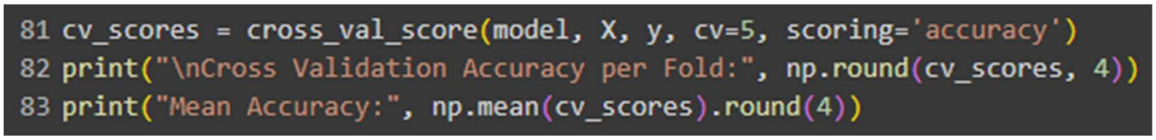
```
print("\nClassification Report:")
print(classification_report(y_test, y_pred, zero_division=0))
```

13. Menampilkan visualisasi confusion matrix dan ROC Curve



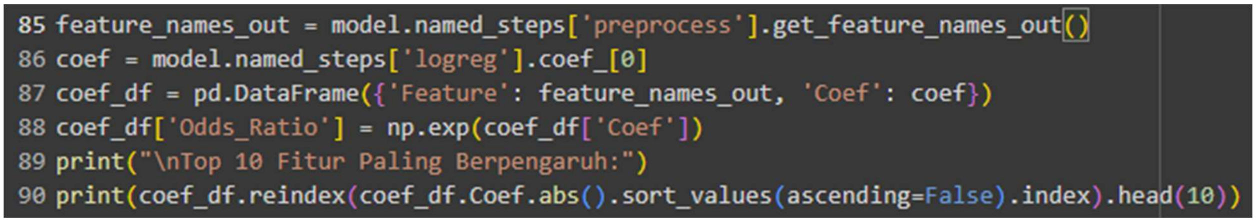
```
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=['Tidak Beli', 'Beli'])
disp.plot()
plt.title("Confusion Matrix Logistic Regression")
plt.show()

RocCurveDisplay.from_estimator(model, X_test, y_test)
plt.title("ROC Curve Logistic Regression")
plt.show()
```

14. Cross validation



```
cv_scores = cross_val_score(model, X, y, cv=5, scoring='accuracy')
print("\nCross Validation Accuracy per Fold:", np.round(cv_scores, 4))
print("Mean Accuracy:", np.mean(cv_scores).round(4))
```

15. Menampilkan fitur paling berpengaruh



```
feature_names_out = model.named_steps['preprocess'].get_feature_names_out()
coef = model.named_steps['logreg'].coef_[0]
coef_df = pd.DataFrame({'Feature': feature_names_out, 'Coef': coef})
coef_df['Odds_Ratio'] = np.exp(coef_df['Coef'])
print("\nTop 10 Fitur Paling Berpengaruh:")
print(coef_df.reindex(coef_df.Coef.abs().sort_values(ascending=False).index).head(10))
```