```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.impute import SimpleImputer
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report

# Load the training dataset
train_data = pd.read_csv('carinsurance_train.csv')

# Preprocessing
numerical_features = ['Age', 'Balance', 'NoOfContacts', 'DaysPassed',
'PrevAttempts']
categorical_features = ['Job', 'Marital', 'Education',
'Communication', 'LastContactMonth', 'Outcome']
binary_features = ['Default', 'HHInsurance', 'CarLoan']

numerical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='mean')),
    ('scaler', StandardScaler())
])
categorical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='most_frequent')),
    ('onehot', OneHotEncoder(handle_unknown='ignore'))
])

preprocessor = ColumnTransformer(
    transformers=[
        ('num', numerical_transformer, numerical_features),
        ('cat', categorical_transformer, categorical_features)
    ],
    remainder='passthrough'
)

# Add Call Duration to training data
train_data['CallStart'] = pd.to_datetime(train_data['CallStart'],
format='%H:%M:%S')
train_data['CallEnd'] = pd.to_datetime(train_data['CallEnd'],
format='%H:%M:%S')
train_data['CallDuration'] = (train_data['CallEnd'] -
train_data['CallStart']).dt.seconds
numerical_features.append('CallDuration')

train_data = train_data.drop(columns=['Id', 'CallStart', 'CallEnd'])

# Separate features and target
X = train_data.drop(columns=['CarInsurance'])
y = train_data['CarInsurance']
```

```
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2,
random_state=42, stratify=y)

# Build model pipeline
model = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('classifier', LogisticRegression(random_state=42, max_iter=500))
])

# Train the model
model.fit(X_train, y_train)

# Evaluate
y_pred = model.predict(X_val)
print(f"Accuracy: {accuracy_score(y_val, y_pred)}")
print(classification_report(y_val, y_pred))

# Load test dataset
test_data = pd.read_csv('carinsurance_test.csv')

# Add Call Duration to test data
test_data['CallStart'] = pd.to_datetime(test_data['CallStart'],
format='%H:%M:%S')
test_data['CallEnd'] = pd.to_datetime(test_data['CallEnd'],
format='%H:%M:%S')
test_data['CallDuration'] = (test_data['CallEnd'] -
test_data['CallStart']).dt.seconds
test_data_processed = test_data.drop(columns=['Id', 'CarInsurance',
'CallStart', 'CallEnd'])

# Predict
predictions = model.predict(test_data_processed)

# Save results
results = test_data[['Id']].copy()
results['CarInsurance'] = predictions
results.to_csv('carinsurance_predictions.csv', index=False)

print("Predictions saved to carinsurance_predictions.csv")
```

```
Accuracy: 0.8075
              precision    recall  f1-score   support

           0       0.82      0.87      0.84       479
           1       0.79      0.71      0.75       321

    accuracy                           0.81       800
   macro avg       0.80      0.79      0.80       800
weighted avg       0.81      0.81      0.81       800
```

```
Predictions saved to carinsurance_predictions.csv
```