```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

# Example data: user_id, action, features, label (1 if action taken, 0 otherwise)
data = pd.DataFrame({
    'user_id': [1, 1, 2, 2, 3, 3],
    'action': ['A', 'B', 'A', 'C', 'B', 'C'],
    'feature1': [0.2, 0.2, 0.5, 0.5, 0.8, 0.8],
    'feature2': [1, 1, 0, 0, 1, 1],
    'taken': [1, 0, 0, 1, 1, 0]
})

# One-hot encode actions
X = pd.get_dummies(data[['action', 'feature1', 'feature2']])
y = data['taken']

# Train/test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train model
model = LogisticRegression()
model.fit(X_train, y_train)

# Predict propensity scores for each action for a new user
new_user = pd.DataFrame({
    'action': ['A', 'B', 'C'],
    'feature1': [0.6, 0.6, 0.6],
    'feature2': [1, 1, 1]
})
new_X = pd.get_dummies(new_user)
# Align columns with training data
new_X = new_X.reindex(columns=X.columns, fill_value=0)

# Get predicted probabilities (propensity scores)
propensity_scores = model.predict_proba(new_X)[:, 1]
new_user['propensity'] = propensity_scores

# Recommend next best action(s)
next_best_action = new_user.sort_values('propensity', ascending=False).iloc[0]
print(f"Next Best Action: {next_best_action['action']} with score {next_best_action['propensity']:.2f}")
```

```
Next Best Action: B with score 0.59
```