로지스틱 회귀를 이용한 클릭 예측 모델

1. 로지스틱 회귀

로지스틱 회귀란?

- 베르누이 분포 : 반응 변수의 확률 범위가 [0,1]
- 이진 분류 : ex) 동전 던지기

로지스틱 회귀의 특징

1. 로지스틱 회귀의 반응 변수의 값 = 긍정 클래스의 확률
- 반응 변수의 값 >= 임계치 : 긍정 클래스를 예측
- 반응 변수의 값 < 임계치 : 부정 클래스를 예측

2. 반응 변수는 로지스틱 함수를 사용해 특징의 선형 조합 함수로 모델링 된다

```
# 로지스틱 함수 = 시그모이드 함수
def sigmoid(z) :
    return 1.0 / (1 + np.exp(-z))
```

2. 온라인 광고 클릭 예측 모델 만들기

```
train_df = pd.read_csv("", nrows = )
unused_columns, label_column = ['id','hour'], 'click'
train_df = train_df.drop(unused_columns, axis = 1)
X_dict_train = list(train_df.drop(label_columns, axis=1).T.to_dict().values())
y_train = train_df[label_column]
```

```
test_df = pd.read_csv("", skiprows = (1,100000) nrows = )
```

```
from sklearn.feature_extraction import DictVectorizer

vectorize = DictVectorize(sparse=True)
X_train = vectorizer.fit_transform(X_dict_train) #적합해서 변환
X_test = vectorizer.transform(X_dict_test) #단순변환
```

```
from sklearn.linear_model.logistic import LogisticRegression

clf = LogisticRegression()
# 매개변수 C : C가 크면 overfitting, 0에 가까워지면 규제가 강해진다
# penalty='l1'(0이된다),'l2'(패널티를 가한다, 계수값이 0에 가까워진다)
clf.fit(X_train, y_train)
```

```
from sklearn.model_selection import GridSearchCV

parameters = {"C" : [0.001,0.01,0.1,1,10], "penalty" : ['l1','l2']}
grid_search = GridSearchCV(clf, parameters, njobs=-1, cv=3, scoring='roc_auc')
grid_search.fit(X_train, y_train)

grid_search.best_params_
```

```
clf_best = grid_search.best_estimator_

y_pred = clf_best.predict(X_test)
y_pred

np.unique(y_pred, return_counts=True)

from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_pred)
```

```
from sklearn.metrics import roc_auc_score, roe_curve
y_pred_proba = clf_best.predict_proba(X_TEST){:, 1} # 두번째 양성확률
y_pred_proba

auc = roc_auc_score(y_test, y_pred_proba)
```