logistic reg hp

March 20, 2024

[1]:

```
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
df=pd.read_csv("C:\\Users\\KENNY\\Desktop\\placement.csv")
df
```

[1]:

```
     cgpa  placement_exam_marks  placed
0    7.19                  26.0       1
1    7.46                  38.0       1
2    7.54                  40.0       1
3    6.42                   8.0       1
4    7.23                  17.0       0
..    …                    …      …
995  8.87                  44.0       1
996  9.12                  65.0       1
997  4.89                  34.0       0
998  8.62                  46.0       1
999  4.90                  10.0       1

[1000 rows x 3 columns]
```

[2]:

```
x=df.drop(['placed'],axis=1)
y=df['placed']
```

[3]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.
 ↪2,random_state=42)
```

[4]:

```
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
x_train_scaled=scaler.fit_transform(x_train)
x_test_scaled=scaler.transform(x_test)
```

[5]:

```
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
model.fit(x_train_scaled,y_train)
y_pred=model.predict(x_test_scaled)
```

```
y_pred
```

[5]:

```
array([0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1,
       0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0,
       1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0,
       1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1,
       0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0,
       0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1,
       0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0,
       1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1,
       0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1,
       0, 1], dtype=int64)
```

[6]:

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

[6]:

```
0.535
```

OPTIMIZED MODEL

[10]:

```
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
df=pd.read_csv("C:\\Users\\KENNY\\Desktop\\placement.csv")
```

[11]:

```
x=df.drop(['placed'],axis=1)
y=df['placed']
```

[12]:

```
from sklearn.model_selection import train_test_split,GridSearchCV
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.
  ↪2,random_state=42)
```

[13]:

```
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
x_train_scaled=scaler.fit_transform(x_train)
x_test_scaled=scaler.transform(x_test)
```

[14]:

```
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
```

[15]:

```
param_grid={
    'penalty':['l1','l2'],
    'C':[0.001,0.01,1,10,100],
    'solver':[ 'liblinear'],
    'dual':[True,False],
    'fit_intercept':[True,False],
    'max_iter':[1000,1500,2000]

}
```

[16]:

```
grid_search=GridSearchCV(model,param_grid,cv=5)
grid_search.fit(x_train,y_train)
```

[16]:

```
GridSearchCV(cv=5, estimator=LogisticRegression(),
             param_grid={'C': [0.001, 0.01, 1, 10, 100], 'dual': [True, False],
                         'fit_intercept': [True, False],
                         'max_iter': [1000, 1500, 2000],
                         'penalty': ['l1', 'l2'], 'solver': ['liblinear']})
```

[17]:

```
best_param=grid_search.best_params_
print('best parameters:',best_param)
```

```
best parameters: {'C': 0.001, 'dual': True, 'fit_intercept': True, 'max_iter':
1000, 'penalty': 'l2', 'solver': 'liblinear'}
```

[18]:

```
best_model=LogisticRegression(**best_param)
best_model.fit(x_train_scaled,y_train)
y_pred=best_model.predict(x_test_scaled)
y_pred
```

[18]:

```
array([0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1,
       0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0,
       1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0,
       1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1,
       0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0,
       0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1,
       0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0,
       1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1,
       0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1,
       0, 1], dtype=int64)
```

[19]:

```
from sklearn.metrics import accuracy_score
Accuracy=accuracy_score(y_test,y_pred)
print('Accuracy:',Accuracy)
```

```
Accuracy: 0.54
```

[ ]:

```

```

[ ]:

```

```