

logistic-regression

December 13, 2023

```
[1]: import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
import time

[2]: data = pd.read_csv(r'C:\Users\user\Desktop\ANJALI RAJ\PERCEPTRON_
↳XOR\WA_Fn-UseC_-Telco-Customer-Churn.csv')

[3]: df = data.copy()

[4]: df['class'] = df['Churn'].apply(lambda x : 1 if x == "Yes" else 0)
# features will be saved as X and our target will be saved as y
X = df[['tenure', 'MonthlyCharges']].copy()
X2 = df[['tenure', 'MonthlyCharges']].copy()
y = df['class'].copy()

[5]: start_time = time.time()
num_iter = 100000
clf = LogisticRegression(fit_intercept=True, max_iter=100000)
clf.fit(df[['tenure', 'MonthlyCharges']], y)
print("Training time (sklearn's LogisticRegression module):" + str(time.time()_
↳ start_time) + " seconds")
print("Learning rate: {} \n Iteration: {}".format(0.1, num_iter))

Training time (sklearn's LogisticRegression module):0.04342818260192871 seconds
Learning rate: 0.1
Iteration: 100000

[6]: result3 = clf.predict(df[['tenure', 'MonthlyCharges']])

[7]: print("Accuracy (sklearn's Logistic Regression):")
f3 = pd.DataFrame(result3).join(y)
f3.loc[f3[0]==f3['class']].shape[0] / f3.shape[0] * 100
```

Accuracy (sklearn's Logistic Regression):

[7]: 78.44668465142695

[]: