Sinhgad Institutes | Date

Name: Pallavi Vijay Patil
Roll No.: COTB028
Subject:

Assignment No. 5

- Aim: Implement logistic regression using python/r to perform classification on - Social_Network_Ads.csv dataset.

- Theory: Logistic Regression: Social Network Ads
This project will be a walkthrough of a simple logistic Regression model in an attempt to strategize a basic ad-targeting campaign for a social media network/website. One of our sponsor's advertisements seems to be particularly successful among our older, wealthier users but seemingly less-so with our younger ones. He'd like to implement an appropriate model. He'd like to show younger ones.

Our dataset contains some info" about all of our users in social networks, including their user ID, Gender, Age and Estimated salary. The last column of the dataset is a vector of booleans describing whether or not each individual ended up clicking on the advertisement (0 = False, 1 = true)

Let's import the relevant libraries, the dataset & establish which variables are either dependent or independent. We'll continually print out any changes that what we've

made to data at bottom of code cells.

```
import numpy as np
import matplotlib.pyplot as plt.
import pandas as pd
dataset = pd.read_csv
dataset.head()
```

If we wanted to determine effect of more independent variables on the outcome, we would have to implement a dimentionally reduction aspect to model because we can only describe so many dimensions visually. However, right now we are only worried about how user's Age and Estimated salary affect their decision to click or not on advertisement. To do this, we will extract relevant vectors (v). The following code segment describes the selection of entire third and fourth columns for x as well as entire fifth column y.

```
x = dataset.iloc [:, [2,3]].values.
y = dataset.iloc [:, 4].values
print (x[3;:])
print ('-' *15)
print (y[:3])
```

```
[[19] 19000]
   [35 20000]
   [26 43000]
   ---  ---.
   [000]
```

Now we need to split our data into two sets: a training set for machine to learn from, as well as a test set for machine to execute on.

This process is reffered to as cross validation and we will be implementing scikit Learn's appropriately named 'train-test-split' class to make it happen. Industry standard usually called for a training set size of 70-80% so we'll split two

```
From sklearn.model_selection import train-test-split
X_train, x_test, y_train, y_test = train_test_split (x, y, test_size = 0.25, random_state = 0)
print (x_train [:3])
print ('-' *15)
print (y_train [:3])
print ('_' *15)
print (y_test [:3])
   [[44 39000]
    [32 120000]
    [38 50000]]
----
[010]
```

To get most accurate results a common tool within ML models is to apply feature scaling: "a method used to standardize range of independ variables of data.

Date

```
In [5]:
print (X_train [:3])
print ('-' *15)
print (X_test [:3])
[[ 0.58164944 -0.88670699]
 [-0.60673761  1.46173768]
 [-0.60673440 -0.36778247]]
---------------
[[-0.80480212  0.50496393]
 [-0.01254409 -0.56778247]
 [-0.30964085  0.15704627]]
```

Now we are ready to build our logistic regression model. We create an object of LogisticRegression() class and refer to it as our 'classifier' for obvious reasons.

- Conclusion:

This confusion matrix tell us that there were 89 correct predictions and 11 incorrect ones, meaning model overall accomplished an 89% accuracy rating. This is good there many ways to improved the model by parameter turning & sample size increasing, but those topics are outside scope of this project.

Our next is to create visualizations to compare the training set & test set. As we've stated through out this discussion, seeing our being able to visualize our work in front of us is imperative to understanding each step of model.