IE 345 - K “Introduction to Deep Learning: Fundamentals Concepts”

Prof. Yuzo

Classification ¶

Logistic Regression

pg. 81 - 84

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [3]:

```
dataset = pd.read_csv('C:/Users/pablo/Desktop/Pablo David/UNICAMP/Python/IE345-K_DeepLe
arning/Social_Network_Ads.csv')
x = dataset.iloc[:, [2, 3]].values
y = dataset.iloc[:, 4].values
print('X: ' ,x)
print('Y: ' ,y)

# Splitting the dataset into the training and test set
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)
```

```
X:  [[    19  19000]
 [    35  20000]
 [    26  43000]
 [    27  57000]
 [    19  76000]
 [    27  58000]
 [    27  84000]
 [    32 150000]
 [    25  33000]
 [    35  65000]]
Y:  [0 0 0 0 0 0 0 1 0 0]
```

In [5]:

```
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)

# Fitting Logistic Regression to the Training set
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0)
classifier.fit(x_train, y_train)

# Predicting the Test set result
y_pred = classifier.predict(x_test)
```

In [6]:

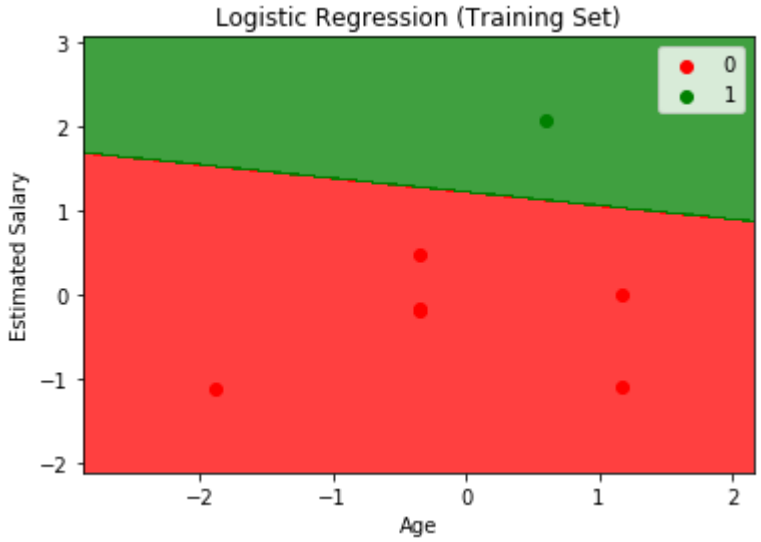
```
# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

In [8]:

```
# Visualising the Training set result
from matplotlib.colors import ListedColormap
x_set, y_set = x_train, y_train
X1, X2 = np.meshgrid(np.arange(start=x_set[:,0].min() - 1,
                               stop=x_set[:,0].max() + 1,
                               step = 0.01),
                     np.arange(start=x_set[:,1].min() - 1,
                               stop=x_set[:,1].max() + 1,
                               step = 0.01))

plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(),X2.ravel()]).T).reshape(X1
.shape),
             alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())

for i, j in enumerate(np.unique(y_set)):
    plt.scatter(x_set[y_set == j,0], x_set[y_set == j,1],
                c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('Logistic Regression (Training Set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
```

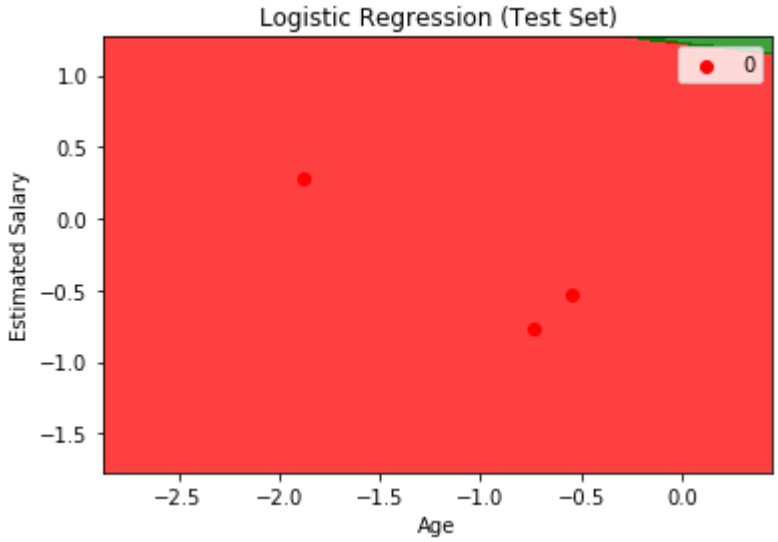


In [9]:

```
# Visualising the Test set result
x_set, y_set = x_test, y_test
X1, X2 = np.meshgrid(np.arange(start=x_set[:,0].min() - 1,
                               stop=x_set[:,0].max() + 1,
                               step = 0.01),
                     np.arange(start=x_set[:,1].min() - 1,
                               stop=x_set[:,1].max() + 1,
                               step = 0.01))

plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(),X2.ravel()]).T).reshape(X1
.shape),
             alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())

for i, j in enumerate(np.unique(y_set)):
    plt.scatter(x_set[y_set == j,0], x_set[y_set == j,1],
                c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('Logistic Regression (Test Set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
```



Pablo David Minango Negrete

pablodavid218@gmail.com