ISCO630E-ASSIGNEMNT-3

Conclusion

By Nairit Banerjee-IIT2016505

Question 1

We need to apply Logistic regression on the **Two exam results dataset** to predict whether a chip gets accepted or rejected.

The regularized cost function is defined as

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log(h_\theta(x^{(i)})) - (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2.$$

We then apply gradient descent as follows

$$\begin{aligned}
&\text{Repeat } \{ \\
&\quad \theta_0 := \theta_0 - \alpha\, \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)} \\
&\quad \theta_j := \theta_j - \alpha\left[\left(\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}\right) + \boxed{\frac{\lambda}{m}\theta_j}\right] \qquad j \in \{1, 2...n\} \\
&\}
\end{aligned}$$

where

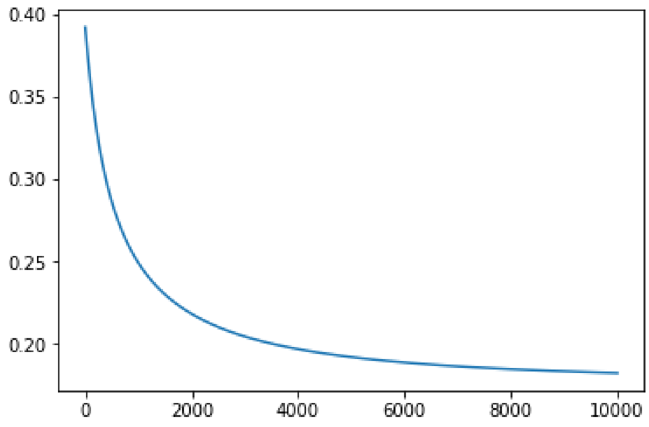
$$h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$$

The data is first normalized and splitted in 70:30 ratio for training and testing respectively.

We apply gradient descent with **learning rate 0.01** and **regularization parameter 0.01**.

The **final value of cost function is 0.182**.

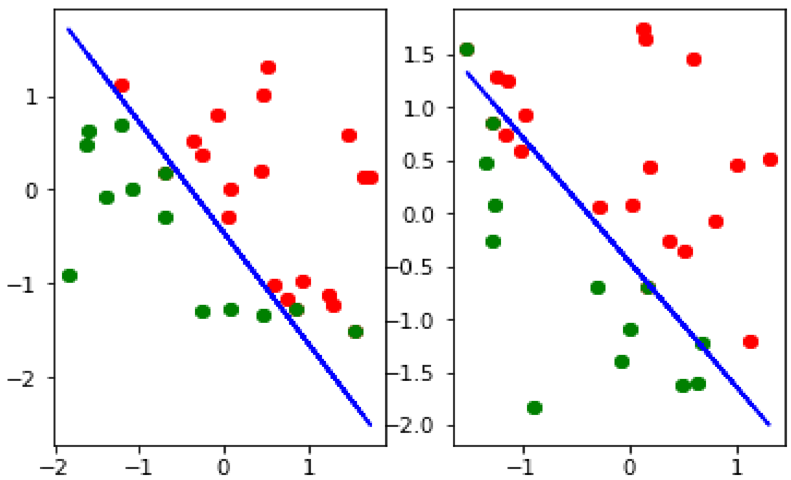
The cost vs iterations graph is as follows



The **final values of the coefficients are : [1.18909504 3.08034019 2.61431181]**

We also got an **accuracy of 90% on the testing data**.

The decision boundaries for both features individually can be visualized as



Question 2

We need to apply Logistic regression on the **Microchip Quality dataset** to predict whether a student gets admitted or not.

The regularized cost function is defined as

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log(h_\theta(x^{(i)})) - (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2.$$

Now we apply Gradient Descent as follows

$$\begin{aligned}
&\text{Repeat } \{\\
&\quad \theta_0 := \theta_0 - \alpha\,\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}\\
&\quad \theta_j := \theta_j - \alpha\left[\left(\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}\right) + \boxed{\frac{\lambda}{m}\theta_j}\right] \qquad j \in \{1, 2...n\}\\
&\}
\end{aligned}$$

where

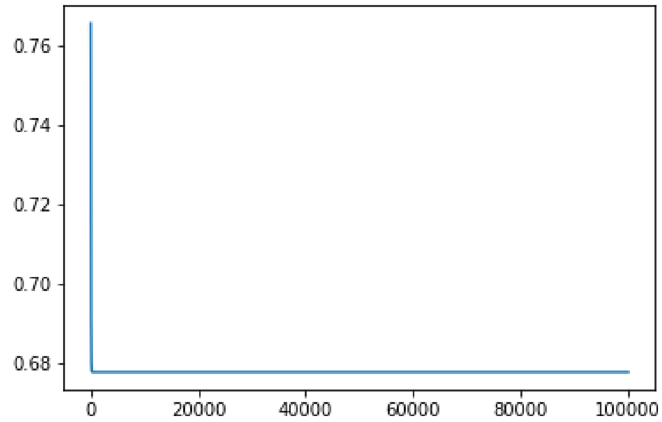
$$h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$$

The data is first normalized and splitted in 70:30 ratio for training and testing respectively.

We apply gradient descent with **learning rate 0.05** and **regularization parameter 0.1.**

The **final value of cost function is 0.677**.

The cost vs iterations graph is as follows,



The **final values of the coefficients are : [-0.15857754 -0.27927304 -0.14269187]**

We also got an **accuracy of 33.33% on the testing data**.

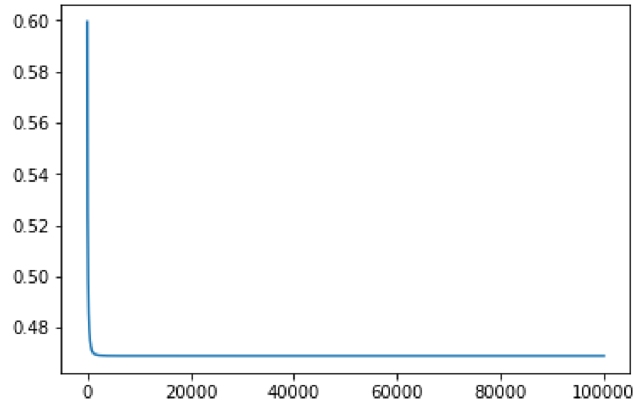
Clearly the model is underfitting the data. Hence we **introduce new features**, which are

X1^2, X1^3, X2^2 and X2^3 where X1 and X2 were our initial features.

We apply gradient descent and final cost came out to be **0.468**.

The coefficients after training were : **[-0.0682349 -1.1433977 -0.34158562 -1.27805802 -1.77472167 0.77207108 1.81546894]**

The cost vs iterations graph is as follows,



Using the same parameters as above, we now get an **accuracy of 83.33%** on the testing split.