

## گزارش کار پروژ ه سوم درس هوش مصنوعی

### اعضای گروه:

احسان حاجیاسینی

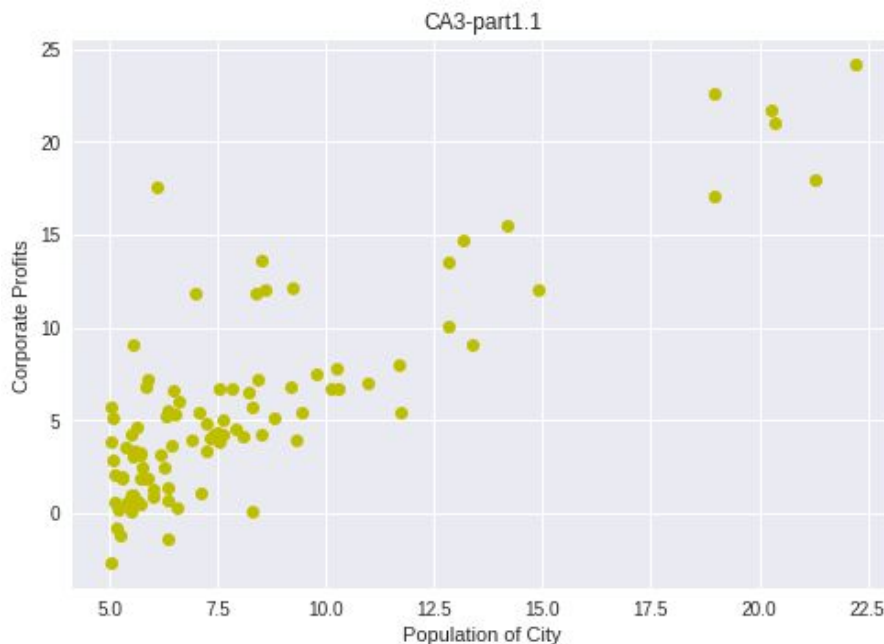
سحر رجبی

مهسا قزوینی نژاد

## ● بخش اول: رگرسیون خطی Linear Regression

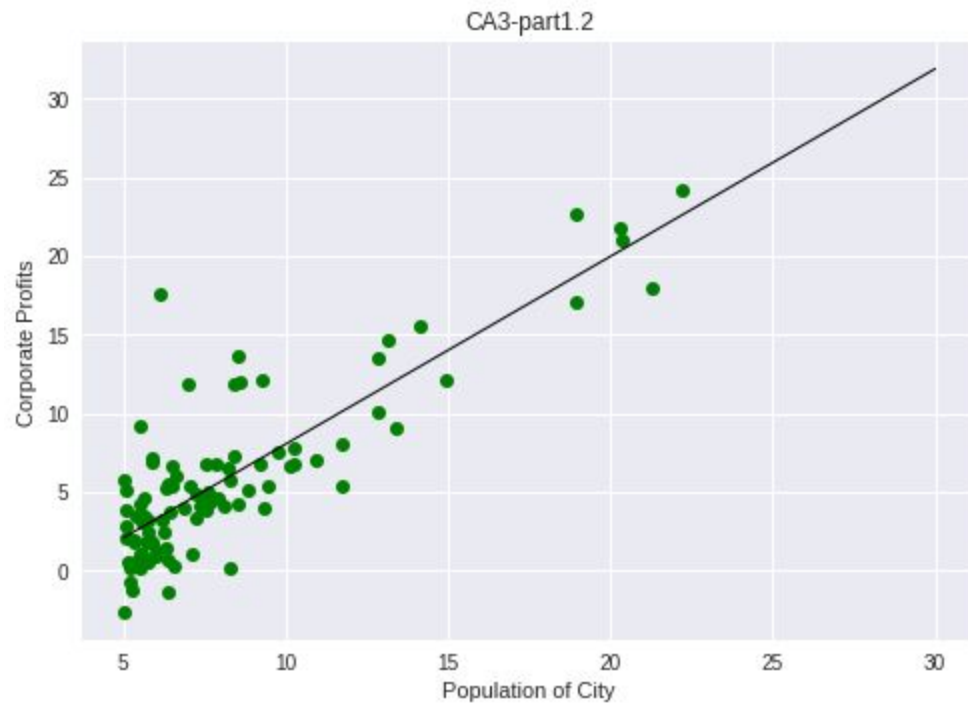
### 1.1 - ترسیم داده - Visualization

قبل از بررسی داده‌ها آن را با استفاده از `matplotlib.pyplot` ترسیم کردیم و به نمودار فوق رسیدیم.

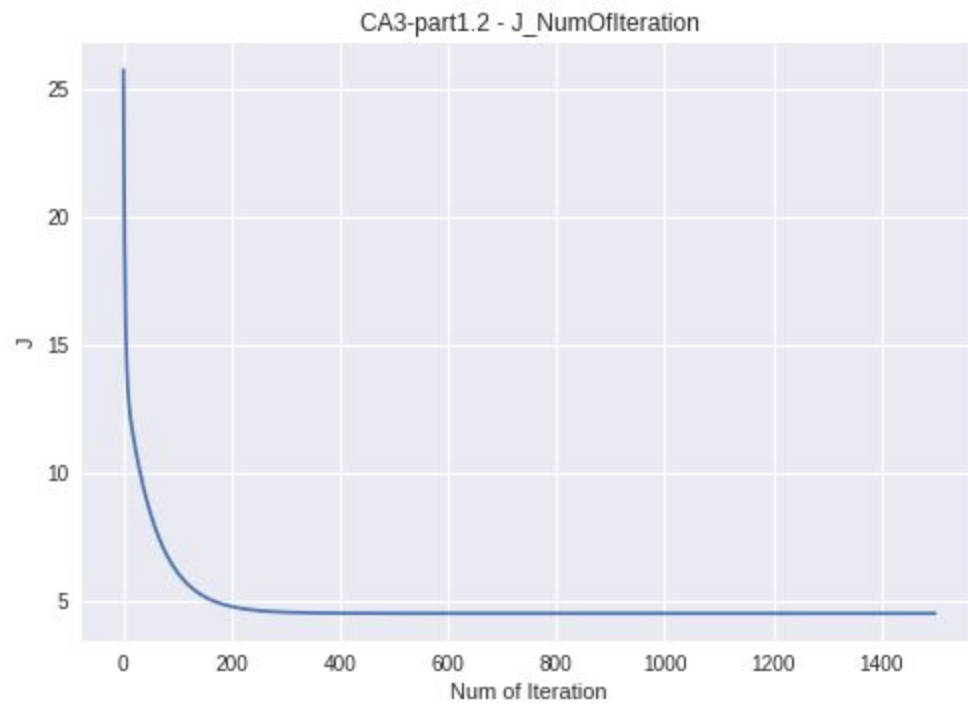


### 1.2.2 - گرادیان کاهشی - Gradient Descent - پیاده‌سازی

مقدار  $\alpha$  را می‌دانیم بین ۰ تا ۱۰ است. به این صورت عمل می‌کنیم که هر بار سقف و کفی برای  $\alpha$  داریم. میانگین این مقادیر را به عنوان  $\alpha$  قرار می‌دهیم. اگر در مرحله‌ای پس از محاسبه‌ی تابع هزینه، دیدیم که هزینه بیشتر از مقدار قبل از آپدیت نتایج آن شد یعنی مقدار  $\alpha$  بزرگتر از چیزی که باید است. بنابراین این سقف جدیدی برای  $\alpha$  به دست آوردیم و دوباره  $\alpha$ های جدید را پیدا می‌کنیم. این کار را به جای این که برای هر دفعه اجرای الگوریتم با داده‌های متفاوت  $\alpha$ های مناسب را بیابیم، انجام می‌دهیم. در نهایت اجرای الگوریتم پس از محاسبه‌ی تنها خط  $y = \theta_1 x + \theta_0$  را نیز به همراه نمودار رسم می‌کنیم تا بتوانیم خروجی خود را ببینیم و شکل به صورت زیر است:

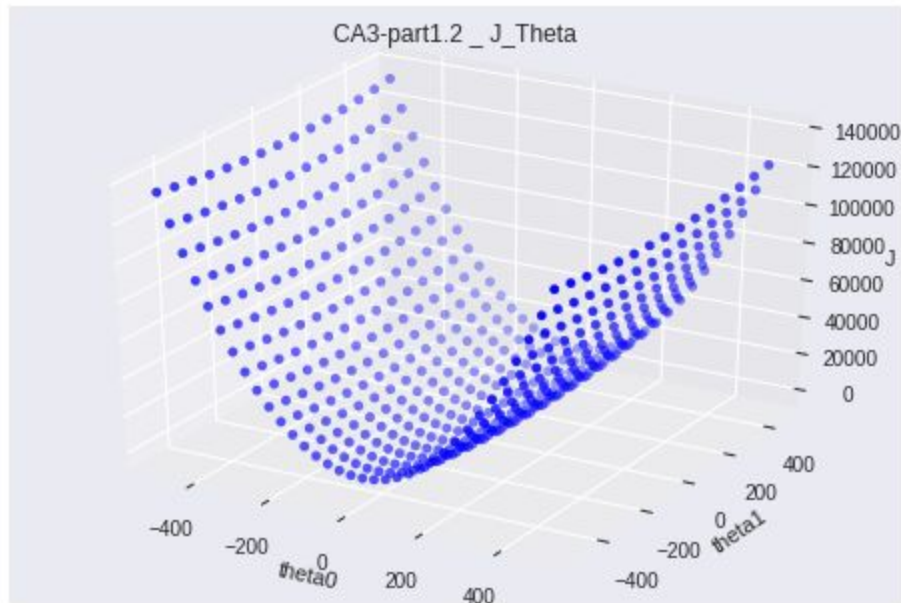


و نیز تابع  $J$  را بر حسب تعداد  $iteration$  ها نمایش می‌دهیم تا نمودار پیشرفت الگوریتم را شاهد باشیم:



### 1.2.3 - گرادیان کاهشی - Gradient Descent - به تصویر کشیدن تابع هزینه (امتیازی)

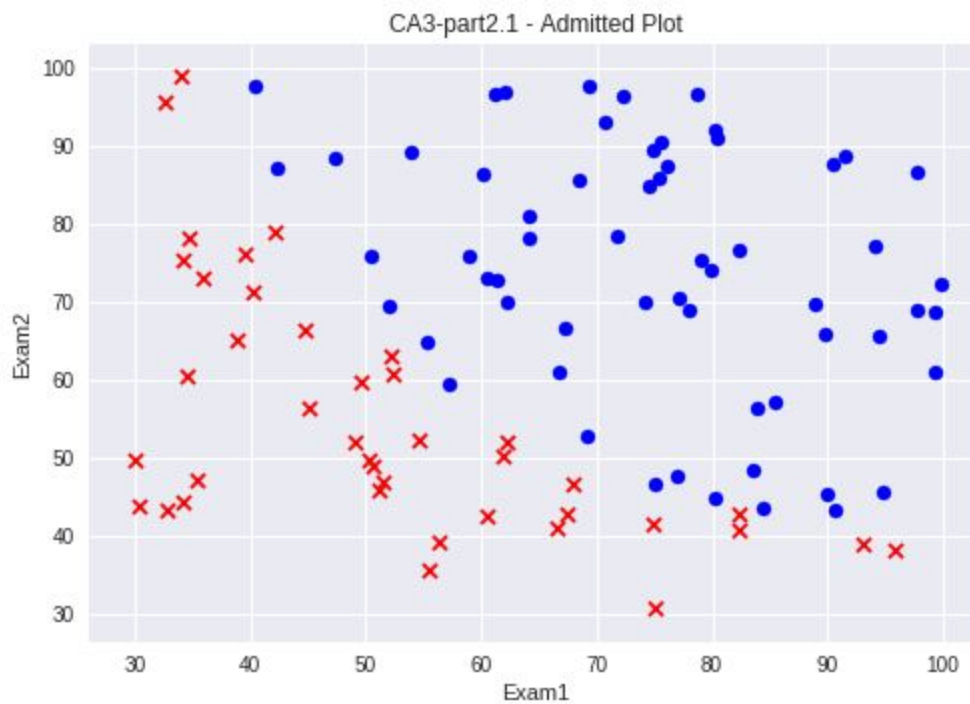
به ازای نتایج مختلف مقدار تابع هزینه یا همان  $J$  را محاسبه می‌کنیم و نمودار آن به شکل زیر است:



### ● بخش دوم: رگرسیون لجستیک Logistic Regression

#### 2.1 - ترسیم داده - Visualization

قبل از شروع پیاده‌سازی الگوریتم داده را رسم کردیم و به نمودار زیر رسیدیم :

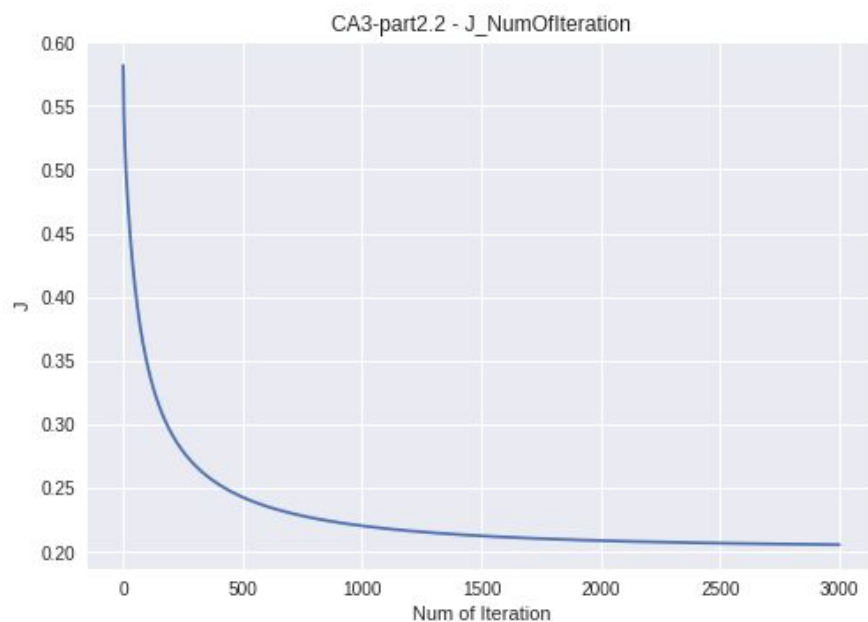


داده‌های آبی به معنی قبولی دانشجو در دانشگاه است و داده‌ی قرمز که با ضربدر مشخص شده‌است به معنی عدم قبولی دانشجو در دانشگاه است.

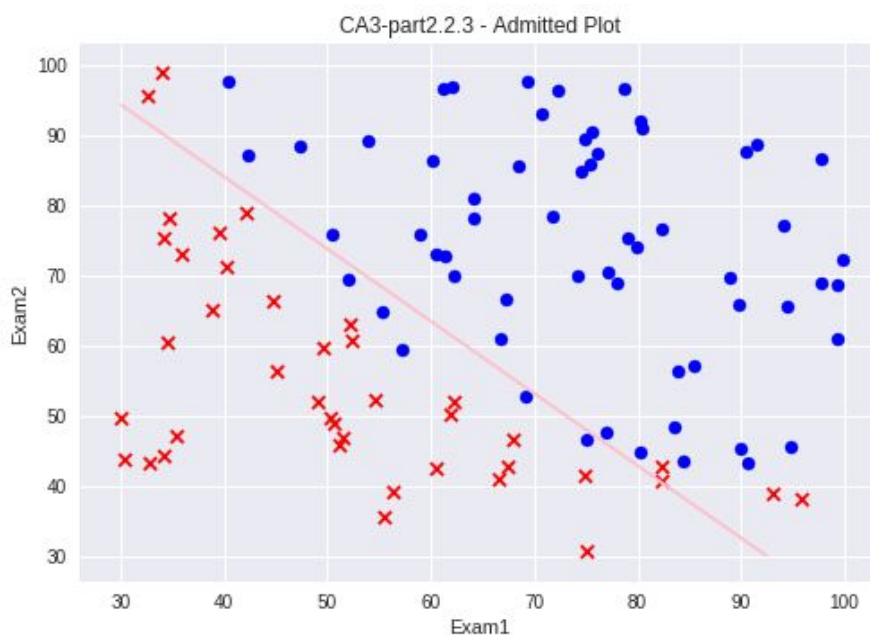
### 2.2.3 - پیاده‌سازی - یادگیری پارامترها

توابع گرادیان کاهشی را مثل سوال قبلی پیاده‌سازی کردیم و از توابع کتابخانه `sklearn` استفاده نکردیم.

مانند قسمت قبل تابع  $J$  را بر حسب تعداد `iteration` ها نمایش می‌دهیم تا نمودار پیشرفت الگوریتم را شاهد باشیم:



پس از یادگیری، تصویر زیر مربوط به داده‌های آموزشی و خط جداکننده آن‌ها است:

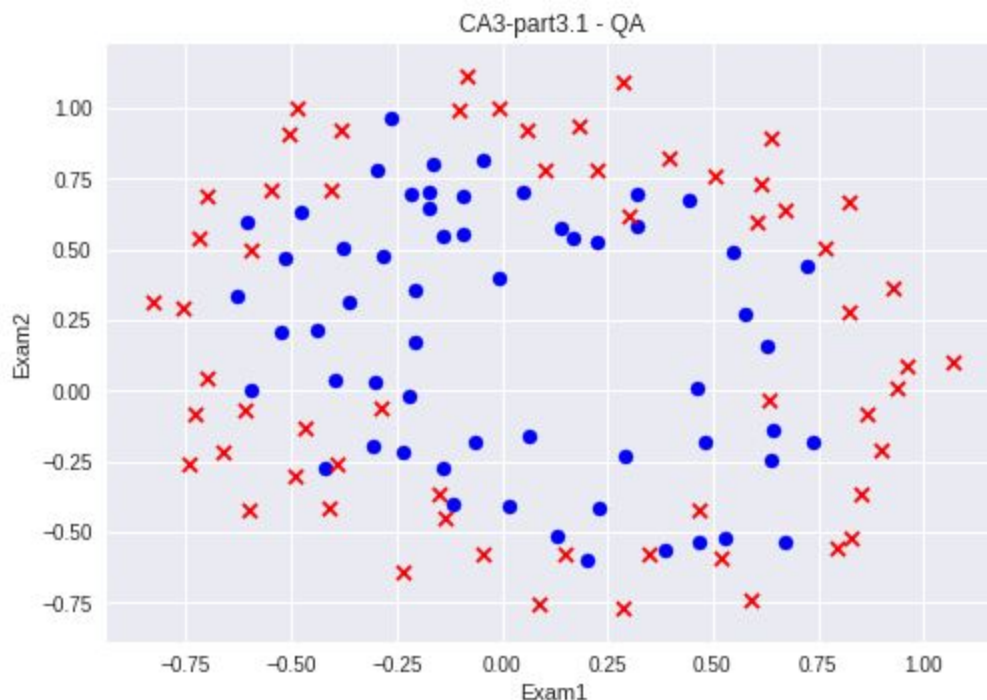


### 2.2.4 - ارزیابی رگرسیون لجستیک

پس از یادگیری پارامترهای مدل، می‌توانیم از آن‌ها برای پیش‌بینی اینکه آیا یک دانشجو پذیرفته خواهد شد یا خیر استفاده کنیم. به عنوان مثال برای یک دانشجو با نمره امتحان ۱، ۴۵ و نمره امتحان ۲، ۸۵ انتظار می‌رود دانشجو پذیرفته شود. ما احتمال پذیرفته شدن این دانشجو را ۰.۷۴۲ محاسبه کرده‌ایم و این مقدار به مقداری که سوال از ما انتظار دارد و ۰.۷۷۶ است بسیار نزدیک است.

## ● بخش سوم: رگرسیون لجستیک قاعدهمند Regularized Logistic Regression

### 3.1 - ترسیم داده - Visualization

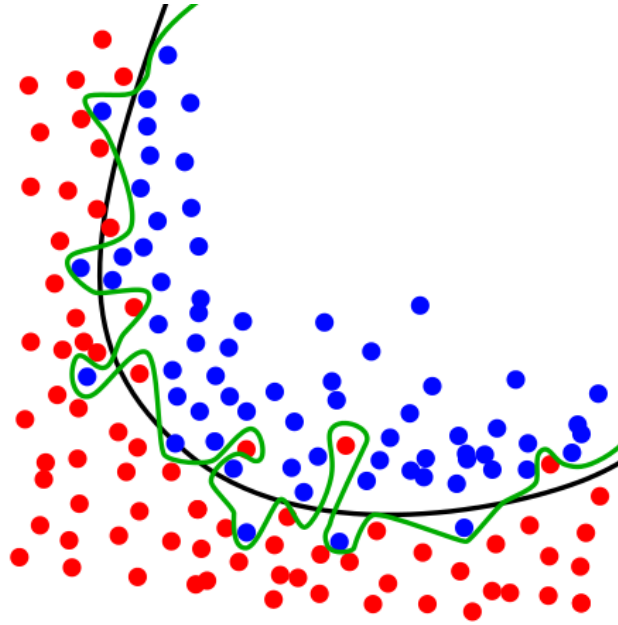


مجموعه داده رسم شده را نمی‌توان به کمک یک خط مستقیم به ۲ گروه تقسیم کرد. پس نمی‌توانیم از قسمت قبل استفاده کنیم.

### 3.2 - نگاشت ویژگی‌ها - Feature Mapping

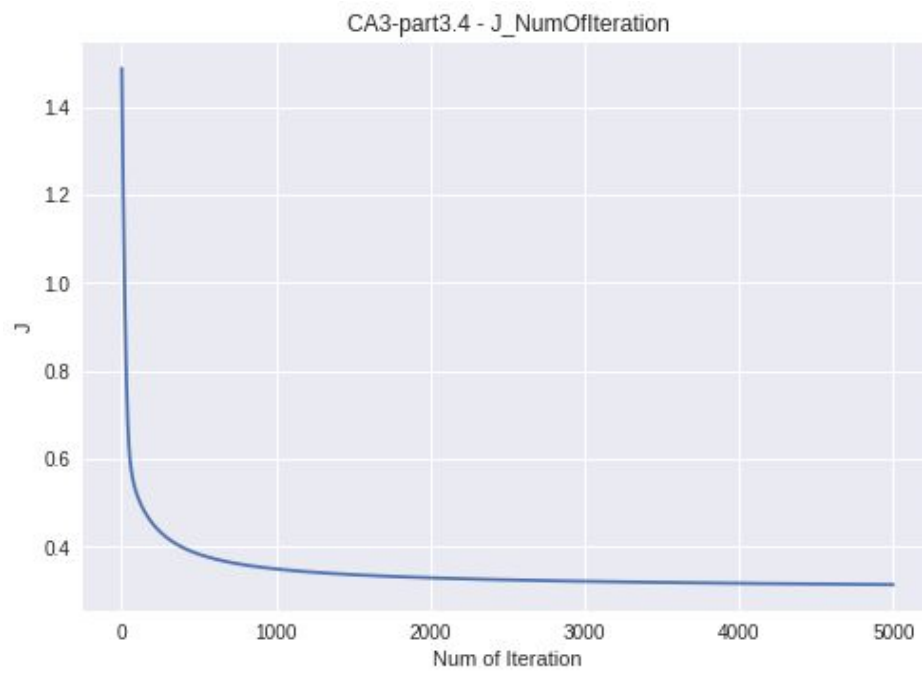
در نتیجه نگاشت گفته شده، بردار ۲ ویژگی به یک بردار ۲۸ بعدی تبدیل می‌شود. رگرسیون لجستیک طبقه‌بندی که با این بردار ویژگی با بعد بالاتر آموزش می‌یابد، مرز تصمیم‌گیری پیچیده‌تری خواهد داشت و زمانی که در یک نمودار دوبعدی ترسیم می‌شود، غیرخطی خواهد بود.

اما در اینجا مشکل انطباق بی‌از حد یا **Overfitting** ممکن است رخ دهد. ممکن است به طور خیلی دقیقی فقط با داده‌های آموزش ما جور شوند و یک شکل دقیقاً دور داده‌های ما بکشد و با این که خطا را واقعاً کاهش می‌دهد اما معتبر نخواهد بود زیرا پیش‌بینی خوبی برای همه‌ی داده‌های ما نخواهد بود  
برای مثال شکل زیر نمونه‌ای از **Overfitting** است:

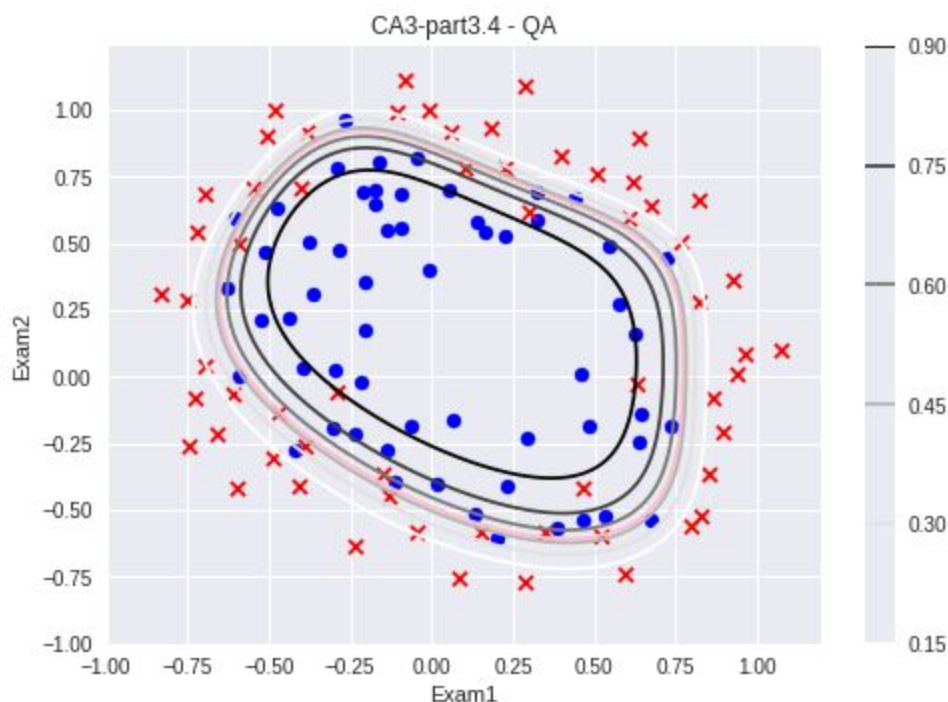


برای حل این مشکل از فرمولی که در قسمت بعد داده شده است استفاده می‌کنیم

مانند قسمت‌های قبل تابع  $J$  را بر حسب تعداد iteration ها نمایش می‌دهیم:



### 3.4 - ترسیم مرز تصمیم‌گیری - Plotting the Decision Boundary امتیازی



خطوط خاکستری تا سیاه همان‌طور که در شکل نشان داده شده است، بیانگر احتمال قابل قبول بودن میکروچیپ است و اگر نقطه داخل خطبسته‌ی صورتی باشد یعنی احتمال قابل قبول بودن آن بیشتر از ۰.۵ است و ما آن را قابل قبول پیش‌بینی می‌کنیم.

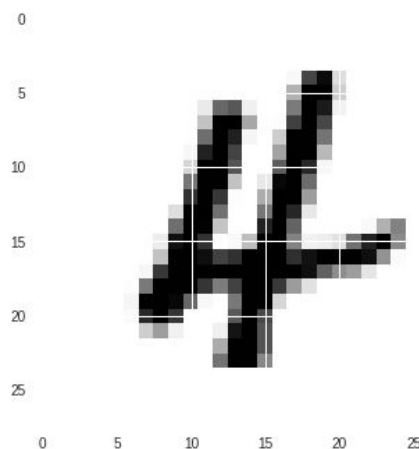
## Multi-class

● بخش چهارم: طبقه‌بند چندکلاسی و شبکه‌های عصبی

## Classification and Neural Networks

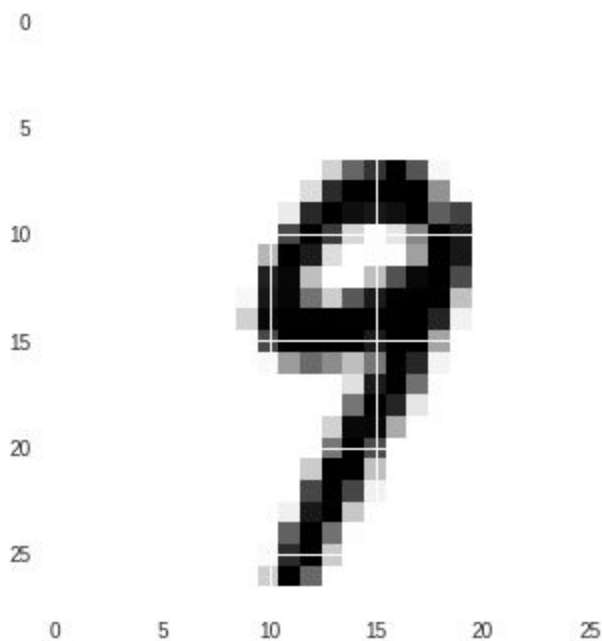
### 4.2 - ترسیم داده‌ها - Visualization the Data

۱۰ سطر از مجموعه داده آموزشی را به صورت تصادفی انتخاب کرده و آن را نمایش دادیم. برای نمونه یکی از آن شکل‌ها به صورت زیر است:



### 4.4.1 - پیش‌بینی یک-در-مقابل-بقیه - One-vs-all Prediction

بعد از آموزش طبقه‌بند خود، می‌توانیم از آن برای پیش‌بینی عدد داخل یک عکس استفاده کنیم. برای هر ورودی احتمال این‌که ورودی متعلق به هر یک از کلاس‌ها باشد را با استفاده از قسمت دوم محاسبه می‌کنیم و محتمل‌ترین را به عنوان پیش‌بینی برای ورودی خروجی می‌دهیم



برای مثال به ازای ورودی بالا خروجی زیر را خواهیم گرفت:

```
predicted probability for value 0 is: 2.02862269536e-06
predicted probability for value 1 is: 1.35786304023e-05
predicted probability for value 2 is: 0.000117601352284
predicted probability for value 3 is: 0.0316158609507
predicted probability for value 4 is: 0.0952218785489
predicted probability for value 5 is: 0.000164973797283
predicted probability for value 6 is: 0.000133520029875
predicted probability for value 7 is: 0.0250409569905
predicted probability for value 8 is: 0.0241093504565
predicted probability for value 9 is: 0.721248613913
```

---

predicted value is: 9

real value is: 9

خطا را به ازای داده‌های تست محاسبه می‌کنیم و مشاهده می‌کنیم که میزان این خطا تقریباً ۹ درصد است.