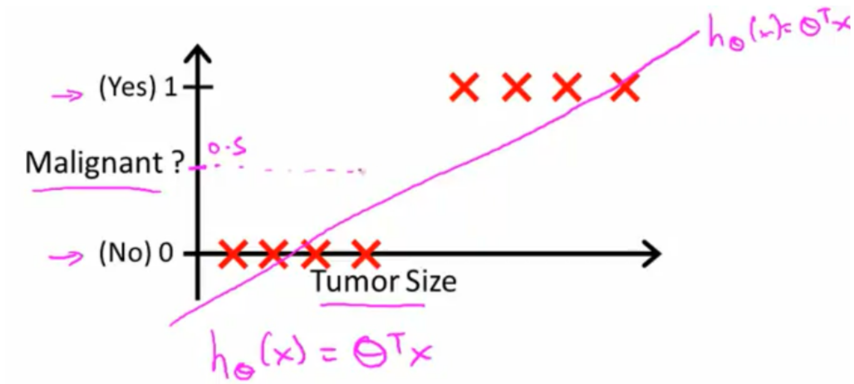
الأسبوع الثالث

Logistic Regression

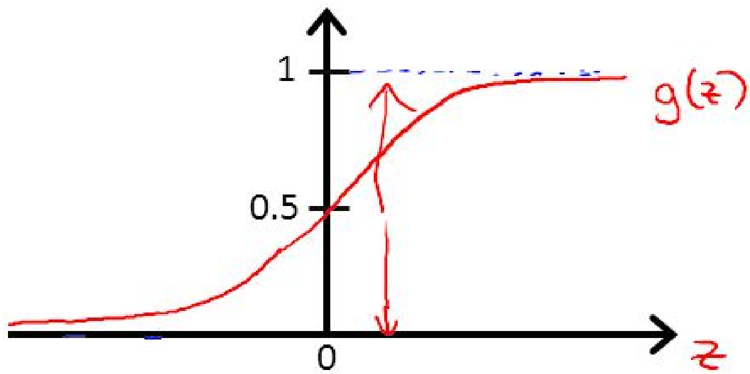
- الكلاسيفيكاشن
  - هل الايميل ده حقيقي ولا سبام
  - هل الورم خبيث ولا حميد
  - هل تحويل الفلوس حقيقي ولا نصب
- غالبا يكون رقم 0 للنيجاتيف , و 1 للبوسيتيف
- اغلب المسائل تكون كلاسين فقط (0 و 1) و بعضها يكون اكثر من كلاس (5,4,3,2,1,0 . . .)



- لو عندي مجموعة نقط هي ورم حميد , ومجموعة تانية ورم خبيث , فلما اعمل خط يجمعهم بيست فيت لاين زي الريجريشن هيعمل لغبطة كبيرة , ومش هيكون عندي امكانية اني اتوقع البيانات بشكل سليم , فلازم اشوف حل تاني
- لازم قيمة الاتش في النهاية تكون بين الصفر و الواحد , مينفعش اكبر من 1 او اقل من صفر ,لان الكلاسيفيكاشن بيقول هي يا صفر يا واحد, عشان كدة هنعمل عملية logistic regression واللي هتحول ارقام النواتج من صفر لواحد
- اسم العملية logistic regression ملوش علاقة بالريجريشن اللي خدناه من شوية , ده كلاسفيكاشن
- ده بيتم عن طريق معادلة اسمها سيجمويد وهي كالتالي

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

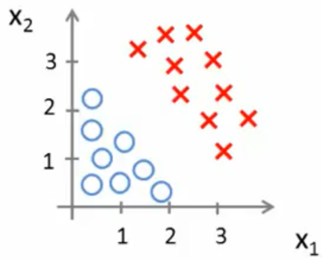
- قيمة ثيتا ترانسبوز في الاكس , اعملها ميناس باور للاكسبونينشيال , بعدها اعمل 1 علي 1 + القيمة ديه
- و هتكون رسمتها كدة



- لاحظ ان الاسمبتوت الاعلي هو 1 و الادني هو صفر, وده معناه ان كل قيم واي تتراوح من صفر لواحد
- طيب هنا سؤال : المفترض ان النتيجة المطلوبة , تقسيم العناصر عندنا لنوعين , ورم حميد (0) او خبيث (1) , طيب لو طلعت النتيجة 0.7 , هنقسمها ازاي ؟
- ده معناها ان احتمالية ان يكون الورم خبيث 70 % , واحتمالية يكون حميد 30 %
- كذلك لو النتيجة 0.24 , يبقي احتمالية خبيث 24 % و حميد 76 %
- فيه اتجاه تاني وهو التقريب , ان لو قيمة الجي طلعت تساوي او اكبر من نص , فده معناها انها واحد (خبيث) لو اقل من نص , يبقي صفر (حميد)

- من الرسم , ومن الاكواشن , واضح ان الجي هتكون 1 , لما قيمة الزي توصل للانفينيتي , عشان e اس سالب انفينيتي هتساوي صفر , ف 1 / 1 هتساوي واحد (النصف الايمن من الرسم)
- وفي حالة الزي تساوي صفر , e اس صفر تساوي 1 ,يبقي 1 / 1 + 1 , هتساوي نص , يعني في النص بالظبط
- في حال زي اقل من صفر , قيم سالبة , هيكون e اس موجب رقم برقم اكبر من 1 يبقي واحد علي رقم اكبر من 2 يبقي اقل من النص
- لما الزي تبقي برقم سالب ضخم جدا (-1000مثلا) e اس موجب الف برقم كبير , يبقي 1 علي الرقم ده برقم يقترب من صفر (النصف الايسر من الرسم)
- لما الزي تكون بسالب انفينيتي , الـ e هتكون اس انفينيتي , يعني انفينيتي , 1 علي انفينيتي تساوي صفر
- يعني من الاخر , علي قد ما زي تكبر , الناتج يساوي 1 , علي قد ما تصغر , يساوي صفر , لو بتساوي صفر , يبقي الجي بنص

- ناخد مثال عملي
    - عندنا الشكل ده , اللي فيه نقط عايزين نقسمهم , و الفنكشن بتاعته معتمدة علي 3 ثيتات
    - هنفرض قيم لثيتا صفر سالب 3 , ثيتا 1 تساوي 1 و ثيتا 2 تساوي 1



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

- فنكشن الاتش (القيمة المتوقعة والمطلوبة , زي السعر المتوقع)اللي عايزينها اكبر من صفر هتكون كدة :

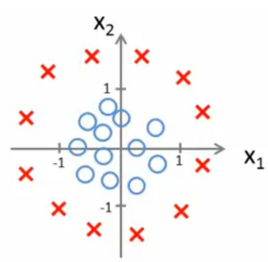
$$-3 + x_1 + x_2 \geq 0$$

- خد بالك ان اكس اتنين كانها واي , يعني ديه معادلة خطية هتمر بكلا من 3 و 3
- و ممكن اقول ان :

X2 > 3 - X1

- وكان كل القيم اللي فوق الخط ده هيكون 1 والقيم اللي تحته هيكون صفر

- مثال تاني , غير خطي
  - الرسم ده :



- مش هينفع نتعامل معاها تعامل الريجريشن الخطي , فمش هينفع نفرض ثيتا 0 و 1 و 2 , عشان نحط اكستين ,, لكن عشان الرسم ده يشابه رسم الدايرة ,فلازم اتعامل مع مربع كلا من اكس 1 و 2 , عشان كدة يكون الفرض فيه 5 ثيتات تغطي اربع اكسات كالتالي :

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$$

- هتكون معادلة تربيعية , للاس التاني , ولو فرضنا ان ثيتا 0 = سالب 1 , و كلا من ثيتا 1 و 2 = 0 , وكلا من ثيتا 3 و 4 = 1 , هتكون كالتالي :

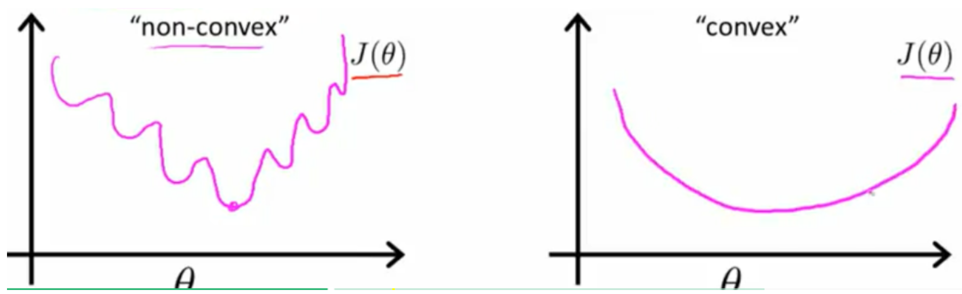
$$x_1^2 + x_2^2 \geq 1$$

- وهي معادلة دائرة , مركزها الاوريجين , ونص قطرها 1 , فاي قيمة ابعد من كدة تكون ايجابي , واي قيمة اصغر من كدة تكون سلبي
- ولاحظ ان وارد ييجي باراميترز اكتر من كدة , ويكون فيه اكسات و ثيتات كتير , فهتيجي اشكال معقدة

- ماذا عن المعادلة المستخدمة
  - مش هينفع نستخدم نفس معادلة الكوست المستخدمة في اللينيار اللي هي :

$$J(\theta)=\frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\left(h_\theta(x^{(i)})-y^{(i)}\right)^2$$

  - ديه مشكلتها انها هتعمل شكل غير متدرج (non convex) زي اللي علي الشمال. واللي هتصعب علينا جدا ايجاد اقل كوست , فلازم نشوف معادلة تانية عشان تعمل لنا شكل convex زي اللي علي الميني

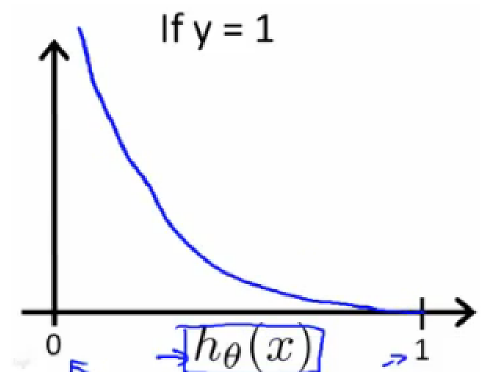


  - فلازم نستخدم اللوجاريثم

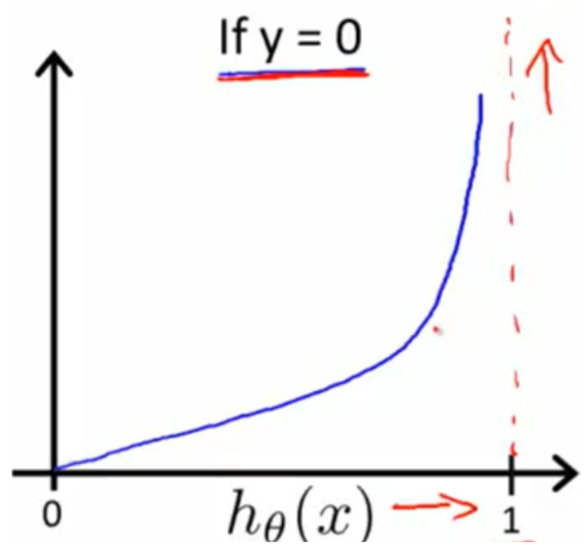
  - المعادلة المستخدمة باللوجاريثم هي :

$$J(\theta)=\frac{1}{m}\sum_{i=1}^{m}\mathrm{Cost}\left(h_\theta\left(x^{(i)}\right),y^{(i)}\right)$$
$$\mathrm{Cost}\left(h_\theta(x),y\right)=-\log\left(h_\theta(x)\right) \qquad \text{if y} = 1$$
$$\mathrm{Cost}\left(h_\theta(x),y\right)=-\log\left(1-h_\theta(x)\right) \qquad \text{if y} = 0$$

  - الأول عايزين نفهم المعني التطبيقي لمعني ان Y تساوي صفر او واحد
    - الـ Y دايما بتكون القيمة الحقيقية المعطاة في الـ Training Data يعني عندي بيانات الف مريض بالسرطان , بياناتهم الصحية ووزنهم وعاداتهم في الاكل و ده ,كل ديه features للـ X
    - بينما النتيجة النهائي , يا تري هل هو مريض او لا هي قيمة الـ Y
    - معني كدة ان كلمة ان Y تساوي صفر معناها بنتكلم عن الناس اللي معندهمش سرطان , او اللي بيتهم متباعش , او اللي منجحش في الامتحان و هكذا, والعكس مع Y تساوي 1
  - في الحالة الأولي : y تساوي 1 هتكون الرسمة كالتالي :



- محور اكس هو الـ h اللي هو القيمة المتوقعة من الاكواشن (يعني سعر البيت المتوقع لمساحة كذا), محور واي هو الـ J اللي هو التكلفة الحقيقية (فرق السعر الحقيقي عن المتوقع سكوير علي ضعف الـ m)
- واضح ان لما كانت الاتش تقترب من 1 (واحنا اصلا في حالة ان الـ y تساوي 1), فالتكلفة (اللي عايزين نقللها للصفر) بتقرب من صفر , ولما الاتش تقترب من صف , فالتكلفة بتزيد بشكل جنوني



- بينما في الحالة التانية (y=0) لما تقل الاتش , الكوست هتوصل لصفر , ولما توصل الاتش لواحد , التكلفة هتوصل لانفينيتي
- وممكن اجمع المعادلتين مع بعض هنا

$$\text{Cost}(h_\theta(x), y) = -y \log(h_\theta(x)) - (1-y)\log(1-h_\theta(x))$$

- لو عوضت عن الواي 1 هتجيبلك المعادلة الاولي , لو عوضت بصفر هتجيبلك التانية , واللي هتأدي للمعادلة ديه

$$\begin{aligned} J(\theta) &= \frac{1}{m}\sum_{i=1}^{m} \text{Cost}(h_\theta(x^{(i)}), y^{(i)}) \\ &= -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log\left(1-h_\theta(x^{(i)})\right)\right] \end{aligned}$$

- نبدا نحاول نجيب قيمة ثيتا اللي هتقلل الـ J , فنقول ان الثيتا هتساوي نفس الثيتا ناقص الفا في تفاضل الـ J , واللي هتكون كدة بعد الاختصار

$$\theta_j := \theta_j - \alpha \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

- واللي هنكررها كذا مرة لغاية لما نجيب قيمة ثيتا اللي هتجيب اقل قيمة لـ J
- و ده هيتعمل مع كل الثيتات , من صفر لاعلي عدد
- و مع ان شكلها مطابق لشكل الفنكشن بتاعت الـ linear regression لكن هنا قيمة الاتش تختلف عن اتش اللينيار
- وممكن نحول المعادلة لمصفوفات تكون بالشكل ده :

$$\theta := \theta - \frac{\alpha}{m} X^T \left(g\left(X\theta \right) - \vec{y} \right)$$

- الطريقة السابقة لحساب التكلفة الدنيا, تسمي الـ Gradient Descent
- لكنها ليست الطريقة الوحيدة , هناك ثلاث طرق اخري وهي :
  - Conjugate Gradient
  - BFGS
  - L-BFGS
- وتلك الطرق هي افضل و اسرع لحساب التكلفة الدنيا , كما انها لا تحتاج الي حساب لقيمة الفا , لكنها اصعب و اكثر تعقيدا
- و لاحظ ان الكود و الخوارزم الخاص بالطرق التلاتة ديه معقد كتير , وكتير من الناس شغالين عليه عادي من غير ما يعرفو آلية الخوارزم ايه

```
function [jVal, gradient] = costFunction(theta)
  jVal = (theta(1)-5)^2 + (theta(2)-5)^2;

gradient = zeros(2,1);
gradient(1) = 2*(theta(1)-5);
gradient(2) = 2*(theta(2)-5);
end
```

- الاول هنا دالة اسمها cost function اللي هيكون فيها مدخل واحد هو الثيتا , ومخرجين الـ jval , gradient

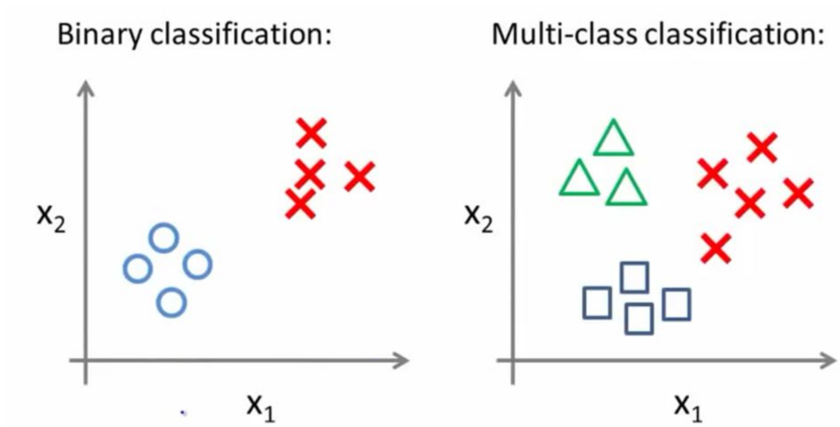
- لاحظ ان اول كلمة فنكشن عشان الاوكتيف يعرف هي ايه , بعدها باكتب الحاجات اللي هتخرج منها , بعدها يساوي , بعدها اسم الدالة اللي هيستخدم , بعدها المدخل (ثيتا)
- قيمة الجي فيل , هتكون بناء علي الفرض الخاص بالثيتا 1 و 2 , ونجيب الجرادينت بنفس الطريقة

- بعدها نروح علي دالة الحساب

```
options = optimset('GradObj', 'on', 'MaxIter', 100);
initialTheta = zeros(2,1);
  [optTheta, functionVal, exitFlag] = fminunc(@costFunction, initialTheta,
options);
```

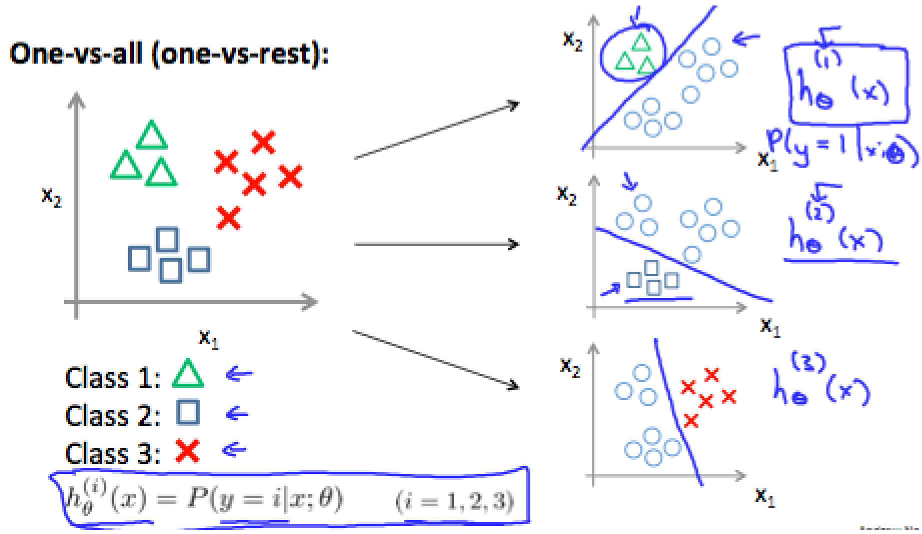
- الاوبشنز : مجموعة من الارقام زي الفيكتور , هتساوي اللي هيطلع من اوبتيمسيت
- اوبتيمسيت , دالة بتعطليها الحاجات ديه , عشان تجيب الاوبشنز
- الثيتات الاولي صفرين
- القيم ديه [optTheta, functionVal, exitFlag] علي الشمال عشان يتحط فيها القيم اللي هتطلع من الفنكشن علي اليمين
- الدالة fminunc وهي المستخدمة بالخوارزمات المعقدة لحساب قيمة الثيتات المستخدمة لحساب اقل قيمة للـ J و يكون مدخلاتها دالة الفنكشن و اليتات و الاوبشنز

- عملية الـ MultiClass Classification
  - معناها ان يتم تقسيم المنتجات الي اقسام عديدة , مش بس 0 و 1 , يعني مثلا ايميل جالك , يا تري هيروح ملف الشغل و لا الاسرة , ولا السفر ولا ايه ؟
  - نفس الموضوع , المريض تشخيصه ايه : مريض بالبرد , بالحمي , بالنقرص , غير مريض
  - و بدل ما كانت في النوع الاول قيمة واي هي 0 و 1 , دلوقتي بقي فيه اكتر من واي

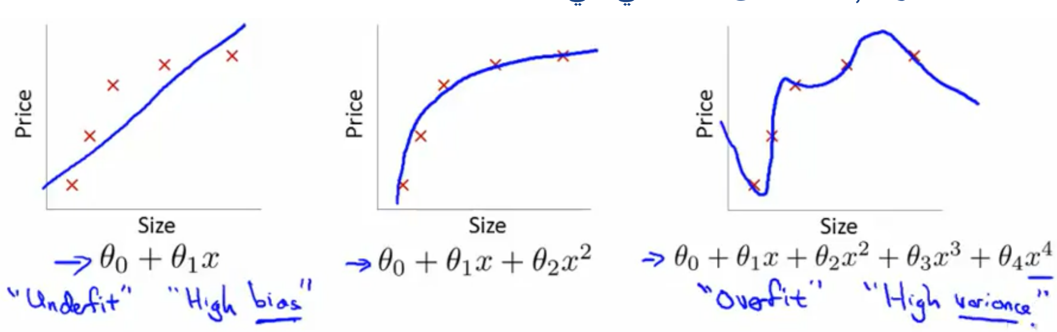


  - و بدل ما كان الرسم زي الشمال يتراوح بين عنصرين , بقي بين اكتر من عنصرين

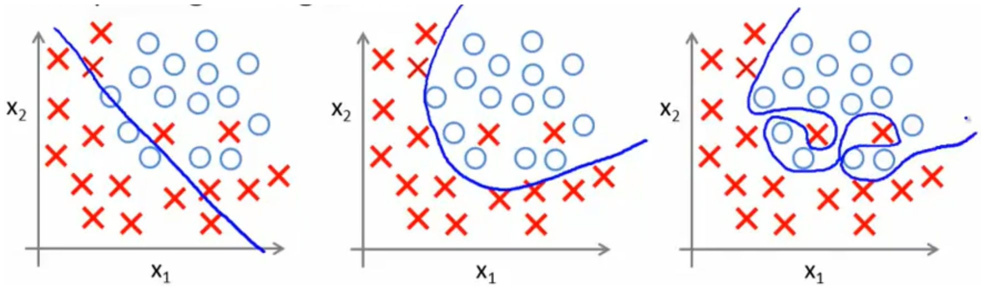
- طب هيتعمل ازاي ؟ ؟ بفرض ان هما 3 اقسام ؟
    - الأول : هنتعامل ان النوع الاول (المثلثات) هي الحاجة المرغوبة (y=1), والباقي غير مرغوبة (y=0) و نعمل خط يخليهم علي جنب , والباقي علي جنب
    - بعدها نكرر نفس الموضوع علي الجزء التاني (الصلبان) , بعدها للتالت (المربعات)
    - بعدها نبدا نحدد القيمة المرغوبة هي فيه الجزء الاول و لا التاني ولا التالت



- مشاكل الـ Overfitting & Underfitting
    - الاشكال التلاتة دول , تمثل نفس النقط هي هي بالظبط



    - الشكل الايسر , نقط متفرقة , فلو عملنا اتنين ثيتا بس , هيعمل خط مستقيم , و ده هيعمل افتقار لدقة البيانات اللي هتخرج , وده اللي اسمه underfitting
    - الشكل الاوسط , تم اختيار 3 ثيتا , هتكون معادلة تربيعية , مظبوطة الي حد كبير , مع ان فيها بعض الاختلاف
    - الشكل الايمن , هو المشكلة , ان الرسم ماشي مظبوط 100 % مع النقط , و ده مشكلة كبيرة ,لان اي قيمة هحاول اتوقعها هتكون ماشية علي النظام القديم و ده معناه ان القيمة المتوقعة مش هتكون سليمة



- هنعرف قدام ادوات , للكشف علي هل فيه OF او UF ولا لا

- طيب علاجه ايه :
  - اما اننا نقلل عدد المعلومات المستخدمة , يعني نقلل عدد العناصر features و نحتفظ بس بالمهم
  - اختيار العناصر و ترك الباقي ممكن يتم يدويا او عبر كود معين اللي يرشح حاجات و يتجاهل حاجات
  - الحل التاني هو الـ regularization يعني ممكن نقول ضبط العناصر , وهي عبر تقليل قيم ثيتات , وده بتستخدم لما باتعامل مع عناصر , تاثيرها في الناتج (سعر البيت) مش كبير ,لكن مينفعش اتجاهله لانه مهم

- طيب عايزين نبدا نعمل الـ Regulization
  - الاول نشوف هل فيه بيانات هيتم استبعادها و احنا شايفين انها بالفعل غير مرتبطة بالناتج ولا لا
  - لو كل البيانات مرتبطة و مش هينفع نستبعد , ممكن نقلل من قيمة بعض البيانات عشان متاثرش كتير
  - فمثلا المعادلة ديه , هيكون فبها اوفر فيتين , ومش عايزين نتخلص من ثيتا 3 و 4 عشان قيم اكس 3 و 4 مهمين , وفي نفس الوقت عايزين نحل الاوفر فيتينج

$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

  - ساعتها لما نعمل معادلة الكوست فنكشن , هنعمل رقم كبير جنب الثيتا 3 و 4 :

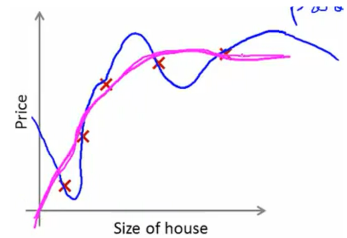
$$min_\theta \ \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + 1000 \cdot \theta_3^2 + 1000 \cdot \theta_4^2$$

  - وجود رقم كبير جنب الثيتات ديه , هتخلي قيم ثيتا 3 و 4 اقل ما يكون , عشان المعامل جنبهم كبير جدا , وده اللي هيادي ان المعادلة الاصلية مش هيكون فيها اوفر فيتينج , لان وجود ثيتا 3 و 4 كانه اختفي

  - طريقة تانية , هي اضيف في معادلة الكوست الـ J , اضيف ليها مجموع كل الثيتات (باستثناء ثيتا صفر , مع ضربها في معامل اسمه لمدا)

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

  - وجود اللمدا اللي بنسميها regulization factor بيعمل تنعيم و ضبط للشكل المتعوج من اللون الازرق اللي هو اوفر فيتينج , للون الاحمر اللي هو سليم



- و متنساش انك لو اخترت لمدا قيمة كبيرة , هتخلي الاكواشن تقلل كل قيم الثيتات (عدا ثيتا صفر اللي مش مجموعة معاهم) لغاية لما توصل قرب الصفر , وده هيخلي ان الإتش (القيمة المتوقعة المطلوبة) هتكون بتساوي ثيتا صفر , ومعناها خط افقي و طبعا ده غلط لانه اندر فيتينج
- فصيغة معادلة الكوست , هتكون كدة بعد اضافة لمدا

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

- و لازم يتم فصل المعادلة لمعادلتين , لان الثيتا صفر مش محسوبة معانا , فالمعادلة الاولي لثيتا صفر من غير اضافة لمدا , و التانية لكل الثيتات , مع اضافة لمدا

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

- هنمسك المعادلة التانية نختصرها شوية هتبقي :

$$\theta_j := \theta_j\,(1 - \alpha\frac{\lambda}{m}) - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})\,x_j^{(i)}$$

- و لاحظ ان كلا من الفا و لمدا و ام ارقام موجبة , معني كدة ان قيمة الفا في لمدا علي ام لازم تكون موجب , يعني 1 ناقص القيمة ديه هيكون اقل من 1 . وغالبا بيكون رقم اقل من 1 بحاجة بسيطة , مثلا 0.99 , فلما تتضرب في ثيتا علي اليمين , بتقلل قيمتها شوية و هكذا لغاية لما توصل لثيتا السليمة

- طيب ماذا عن طريقة النورمال ؟ اللي هي نعمل تفاضل للمعادلة و نساويها بالصفر فورا ؟
  - طريقة الليניار , كانت المعادلة كدة زي ما هي موجودة فوق

$$\theta = (X^TX)^{-1}X^Ty$$

  - دلوقتي عشان فيه موضوع اللمدا , فهيحصل تغيير انها هتكون كدة

$$\theta = \left(X^T X + \lambda \begin{bmatrix} 0 & & & \\ & 1 & & \\ & & 1 & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} \right)^{-1} X^T y$$

- مع اعتبار ان الماتريكس اللي جنب اللمدا , هي ماتريكس مربعة , طولها و عرضها , يساوي n+1 و كل عناصرها صفر , عدا المحور كله وحايد , لكن اول صف و عمود برضه صفر (كأنها مصفوفة Identity بس اول قيمة صفر)
- يعني لو عدد العناصر تلاتة , هتكون المصفوفة :

Matrix A:

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- وخد بالك ان من عيوب طريقة النورمال ان احيانا الماتريكس الداخلية تكون سينجل , يعني مش هيتعملها انفرس , لكن من حسن الحظ ان اضافة الماتريكس الجديدة مضروبة في لمدا , هتمنعها من انها تكون سنجل

- وختاما , هندخل في الفنكشن المهمة : fminunc

fminunc(@costFunction...)

$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}$ theta(1) theta(2) theta(n+1)

Advanced optimization

```
function [jVal, gradient] = costFunction(theta)
    jVal = [code to compute J(θ)];
```

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

gradient(1) = [code to compute $\boxed{\frac{\partial}{\partial \theta_0} J(\theta)}$];

$$\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

gradient(2) = [code to compute $\frac{\partial}{\partial \theta_1} J(\theta)$];

$$\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_1^{(i)} + \frac{\lambda}{m} \theta_1$$

gradient(3) = [code to compute $\frac{\partial}{\partial \theta_2} J(\theta)$];

$$\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_2^{(i)} + \frac{\lambda}{m} \theta_2$$

$\vdots$

gradient(n+1) = [code to compute $\frac{\partial}{\partial \theta_n} J(\theta)$];

- الفنكشن fminunc عشان تشتغل , لازم نحدد كود الفنكشن cost function اللي هي بتستخدمها معاها
- الفنكشن مكتوبة فوق ان هي بتستخدم معلومتين عشان تشتغل : jval & gradient
- الـ jval هي لحساب قيمة الـ j و هي عن طريق الدالة اللوغاريتمية , مضاف اليها قيمة اللمدا الجديدة
- الـ gradient (وهي قيمة الثيتات) , تتقسم لكذا حاجة

- ثيتا صفر , وهي لا تتداخل مع اللمدا , فتتكتب بالشكل اللي هنا
- ثيتا 1 , وهي نفس صيغة ثيتا صفر مضاف اليها قيمة اللمدا
- باقي الثيتات زي ثيتا 1

- متنساش ان كل دوال الجرادينت , هي عبارة عن تفاضل جزئي لدوال الـ J بالنسبة لثيتا المطلوبة

- و اخيرا , دالة الكوست قبل الـ regulation هيكون شكلها كدة

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

- وبعد الريجيولاشن هتكون كدة

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

- الاختبار الثاني :
  - الفكرة :
    - توجد نتائج اختبارين , وبعدها رقم يوضح هل تم قبوله بالجامعة ام لا , الرقمين الاولين هما x1 , x2, ورقم القبول هو y والذي هو 1 او 0
    - نبدا في الشاشة الرئيسية , حيث يتم تحميل البيانات

Ex2()	
data = load('ex2data1.txt');	تحميل البيانات في مصفوفة اسمها داتا 100 ص - 3 ع
X = data(:, [1, 2]);	تحديد مصفوفة اكس بكل صفوف الداتا والعمودين الاول والثاني 100ص-2ع
y = data(:, 3);	مصفوفة واي بالعمود الثالث فقط 100ص - 1ع
plotData(X, y);	الامر بالرسم
PlotData()	
figure; hold on; pos = find(y==1); neg = find(y == 0); plot(X(pos, 1), X(pos, 2), 'k+','LineWidth', 2, 'MarkerSize',	الأمر بالرسم اولا تعيين مين الموجب و مين السالب

7); plot(X(neg, 1), X(neg, 2), 'ko', 'MarkerFaceColor', 'y', 'MarkerSize', 7);	بعدها تحديد رسم النقط الموجبة بشكل معين و السالبة بشكل اخر
هنروح مؤقتا للدالة سيجمويد نفهمها	
Sigmoid() $g(z) = \frac{1}{1 + e^{-z}}.$	
function g = sigmoid(z)	الداخل و الخارج منها هي مصفوفة بنفس مقاس z اللي هي هتكون بنفس مقاس اكس الجديدة 100ص - 3 ع
g = zeros(size(z));	وضع قيم لـ g وهي مصفوفة مطابقة للمدخل 100ص - 3 ع
A1 = e .^ (-z);	ايه 1 مصفوفة بنفس القياس 100-3 , وهي قيمة اكسبونينشيال اس ناقص زي
A2 = ones(size(z)) + A1;	ايه 2 : مصفوفة بنفس القياس 100-3 , وهي اضافة وحايد للايه 1
A3 = 1./A2;	ايه 3 : نفس القياس 100-3 وهي مقلوب كل الارقام في ايه 2
g = A3;	نضع ايه 3 في المخرج
one = ones((size(z))); **g = one ./ (one + e .^ (-z));**	**بديل : يمكن ان نضع السطرين دول مكانهم**
Ex2()	
hold on; xlabel('Exam 1 score') ylabel('Exam 2 score') legend('Admitted', 'Not admitted') hold off; fprintf('\nProgram paused. Press enter to continue.\n'); pause;	لعرض اسماء الصفوف و الاعمدة في الرسم
[m, n] = size(X);	تحديد ابعاد الاكس m=100 n =2
X = [ones(m, 1) X];	اضافة عمود جديد شمال الاكس فيه الوحايد (عشان ثيتا 0)

	اكس بقت 100 ص - 3 ع
initial_theta = zeros(n + 1, 1);	تحديد قيم اوليه لثيتا 3 ص - 1 ع
[cost, grad] = costFunction(initial_theta, X, y);	قيمتين واحدة للتكلفة و التانية للثيتات , وهتيجي من الدالة كوست فنكشن مدخلاتها هي : الثيتات الاولي اصفار (3 ص - 1 ع) إكس (100 ص - 3 ع) واي (100ص - 1ع)
costfunction() $J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log(h_\theta(x^{(i)})) - (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right],$	
one = ones(m,1);	تحديد صفوفة وحايد , 100 صف في عمود واحد
g = sigmoid(X * theta);	السيجمويد هتخرج لنا مصفوفة 100 صف في عمود واحد , و اللي هيتحط اس للاكبسونينشيال لازم يكون محصرة ضرب الاكسات و الثيتات , فهنضرب مصفوفة اكس (100-3) في مصفوفة ثيتا (3-1) , هتكون (100-1) و السيجمويند هتجيبلنا قيمهم اللي تتراوح بين 0 و 1 , في مصفوفة طويلة 100 في 1
J = 1/m * (((log(g))' * -y) - ((log((one - g)))' * (one - y)));	عشان نجيب قيمة لوج السيجمويند (100-1) مضروب في واي , نعمل للسيجمويند ترانسبوز (1-100) ونضربها في الواي (100-1) تعمل لنا قيمة مباشرة لحاصل ضربهم , ومننساش السالب
	الجزء التاني , نجيب لوج السيجمويند بعد ما اطرح منها واحايد (100-1) , ونعمل لها ترانزبوس (1-100) , وبعدها نضربها في الوايات لما نطرح منها واحايد (100-1)
grad = 1/m * X' * (sigmoid(X * theta) - y); $\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$	هجيب السيجمويند (100-1) و اطرح منها الوايات , بعدها تتضرب في ترانسبوز الاكس يعني تراسنبوز الاكس (3-100) في (100-1) , يعملي (3-1) و اقسمها علي الام

<table>
<tr><th colspan="2">Ex2()</th></tr>
<tr><td>fprintf('Cost at initial theta (zeros): %f\n', cost);
fprintf('Expected cost (approx): 0.693\n');
fprintf('Gradient at initial theta (zeros): \n');
fprintf(' %f \n', grad);
fprintf('Expected gradients (approx):\n -0.1000\n -12.0092\n -11.2628\n');

test_theta = [-24; 0.2; 0.2];
[cost, grad] = costFunction(test_theta, X, y);

fprintf('\nCost at test theta: %f\n', cost);
fprintf('Expected cost (approx): 0.218\n');
fprintf('Gradient at test theta: \n');
fprintf(' %f \n', grad);
fprintf('Expected gradients (approx):\n 0.043\n 2.566\n 2.647\n');
fprintf('\nProgram paused. Press enter to continue.\n');
pause;</td><td>عرض البيانات , وقيم الجي و الثيتات في حالتين</td></tr>
<tr><th colspan="2">الدالة fminunc</th></tr>
<tr><td colspan="2">وهي دالة بديلة للي عملناه فوق , لان هي بلت ان في الاوكتيف
و هي محتاجة كذا حاجة عشان تشتغل :
• المعلومات اللي طالعة من كوست فنكشن (كوست , جريد)
• القيم الاولي لثيتا
• معلومات الابوشنز</td></tr>
<tr><td>options = optimset('GradObj', 'on', 'MaxIter', 400);</td><td>الاوبشنز , فيكتور من كذا معلومة , وتم اعلامه أن خيار جريد شغال (عشان يرجع قيم) , وان اقصي عدد للمحاولات 400</td></tr>
<tr><td>[theta, cost] = fminunc(@(t)(costFunction(t, X, y)), initial_theta, options);</td><td>هنا اكتب المخرجت الاول (ثيتا , الكوست) , يساوي دالة fminunc بعدها انادي علي الدالة الكوست , مع قيم ثيتا , والاوبشنز اللي فوق</td></tr>
<tr><td></td><td></td></tr>
</table>