PRÁCTICA 3 – REGRESIÓN LOGÍSTICA

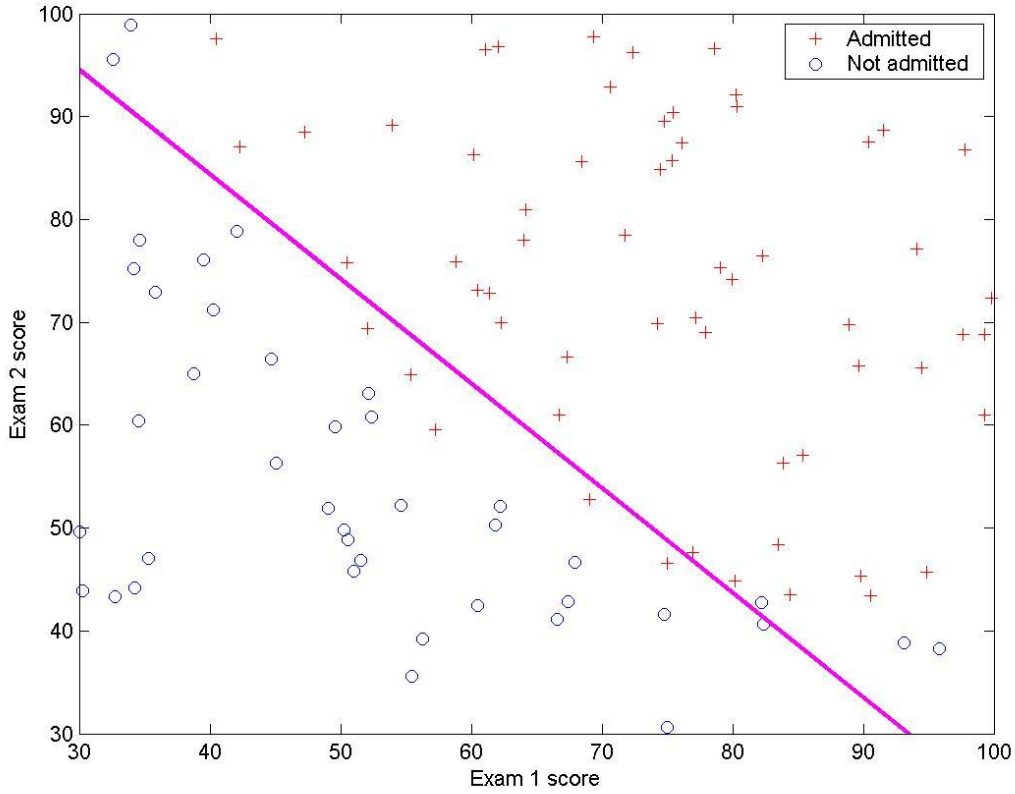
Fernando Aliaga Ramón - 610610

2. Regresión logística básica

Para realizar la regresión logística básica primero hemos calculado los valores de theta.

Valores de theta obtenidos:
-27.071245
0.220119
0.216532

La regresión se puede observar de forma gráfica junto con los datos dados.



Dados estos valores de theta calculamos las predicciones con los datos de entrenamiento y de test. Para realizar dicha predicción hemos definido una función auxiliar 'prediccion' que devuelve un vector con la predicción de la salida y.

```
function h = prediccion(theta, X)

        h = 1./(1+exp(-(X*theta)));

  index_1 = find(h >= 0.5);
        index_0 = find(h < 0.5);

        h(index_1) = ones(size(index_1));
        h(index_0) = zeros(size(index_0));
end
```

Teniendo la predicción de salida sólo tenemos que compararlos con los datos dados.

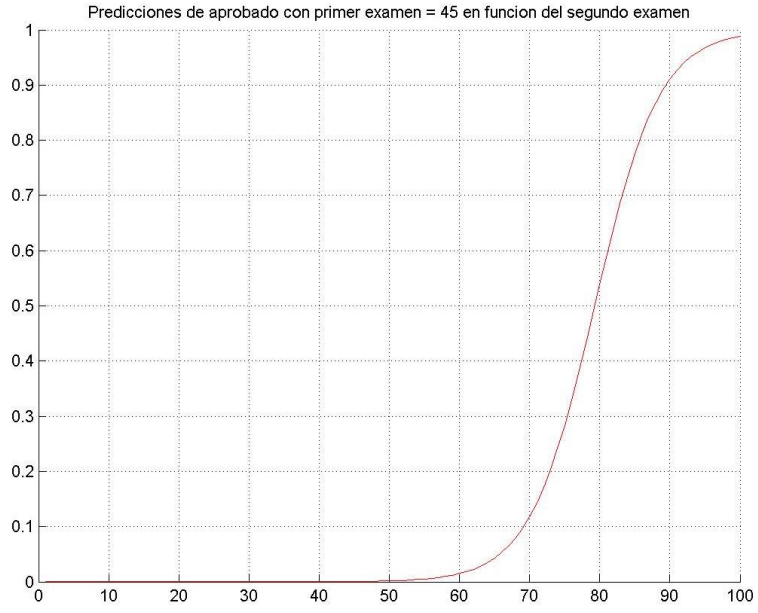
```
Porcentaje de errores = (1 - (mean(double(h == ytr))))*100
```

Dicho porcentajes arrojan los siguientes resultados.

```
Porcentaje de errores con datos de entrenamiento: 8.750000
Porcentaje de errores con datos de test: 15.000000
```

Obviamente, la regresión se comporta mejor ante los datos con la cual ha sido entrenada.

Debemos dibujar una gráfica que muestre la probabilidad de ser admitido si la calificación del primer examen es 45. Para la realización de la misma iteramos desde 0 a 100, calculando en cada iteración el valor de h(x). Podemos ahora ver gráficamente la evolución de la probabilidad.



3. Regularización

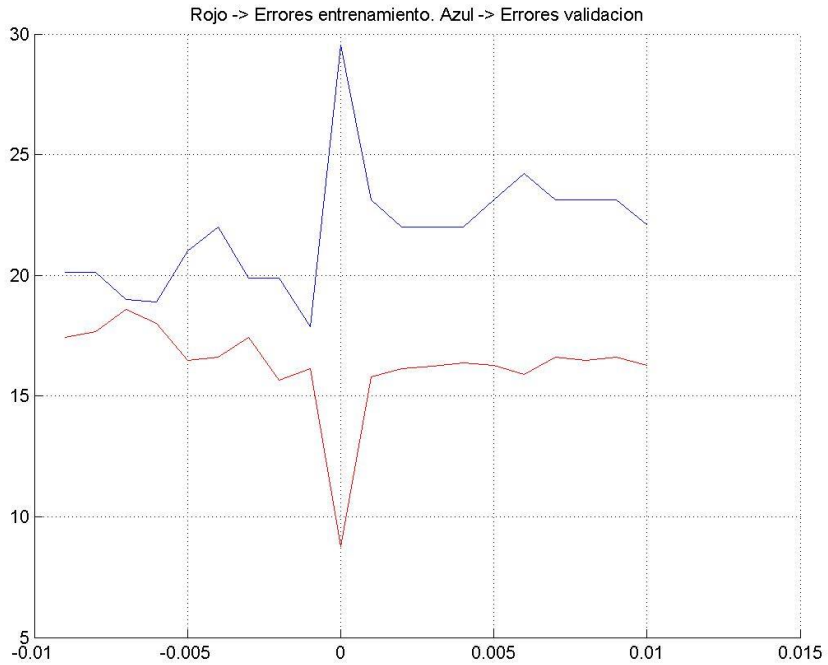
Para usar una regresión logística regularizada debemos escoger un valor de lambda. Dicho valor lo hemos escogido mediante validación cruzada.

La implementación de dicha validación cruzada tiene como claves los siguientes aspectos:

- Los valores de lambda se han buscado en el intervalo de -0.01 a 0.01 con pasos de 0.001.
- El valor de 'k' ha sido definido como 10.
- Para encontrar el mejor valor de lambda hemos tomado como referencia el porcentaje de errores respecto de los datos de validación. Dicho porcentaje se ha obtenido de la misma forma que en el apartado 2, haciendo uso de la predicción y comparando con los datos de salida.
- Los valores de los errores respecto de los datos de entrenamiento y de validación se almacenarán en dos vectores para ser mostrados en una gráfica comparativa.

La ejecución de validación cruzada para la selección de lambda genera los siguientes resultados, los cuales varían en las distintas ejecuciones debido a que los valores iniciales son elegidos aleatoriamente al principio de la ejecución.

```
Fin validacion cruzada.
Mejor lambda : -0.001000
Error con datos de validacion : 17.888889
```



La gráfica representa en el eje X los distintos valores que toma lambda y en el eje Y el porcentaje de errores. La línea de color azul representa los errores con los datos de validación, mientras que la línea de color rojo representa los errores con los datos de entrenamiento.

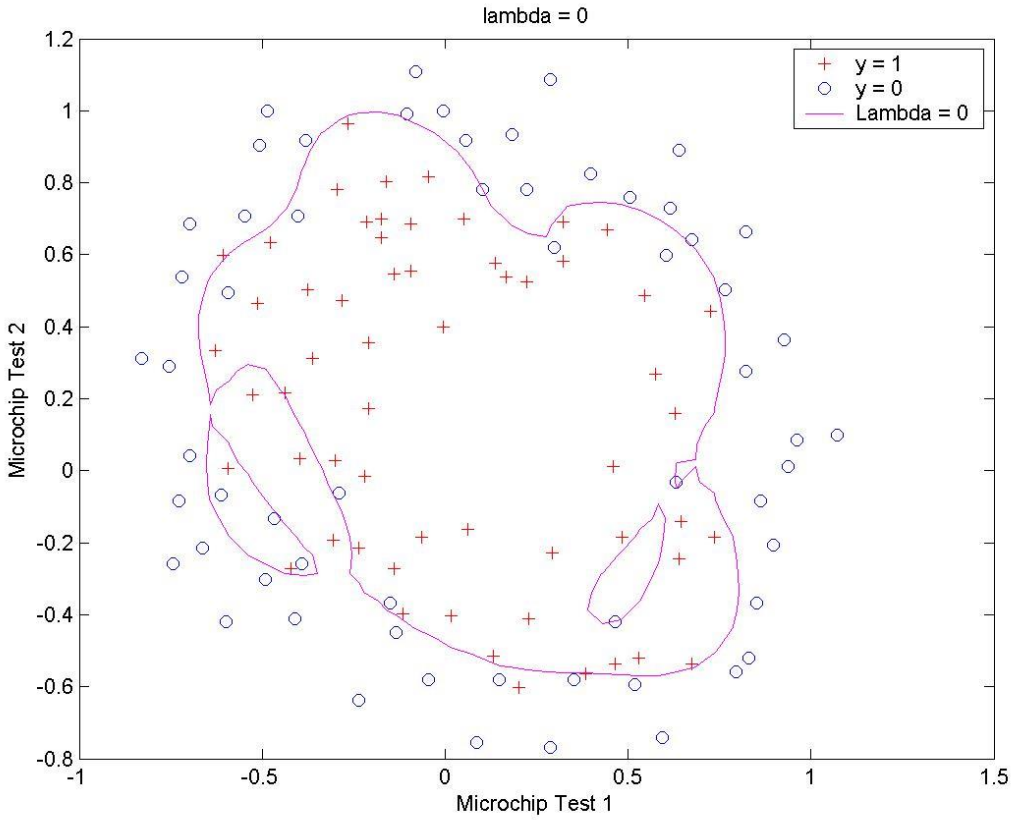
Para determinar el mejor lambda escogemos el valor que hace minimizar el porcentaje de errores con datos de validación, en este caso lambda = -0.001000.

Obtenido el parámetro lambda por validación cruzada, vamos a compararlo con el valor de lambda=0. Para ello vamos a entrenar con los datos de entrenamiento ambas regresiones.

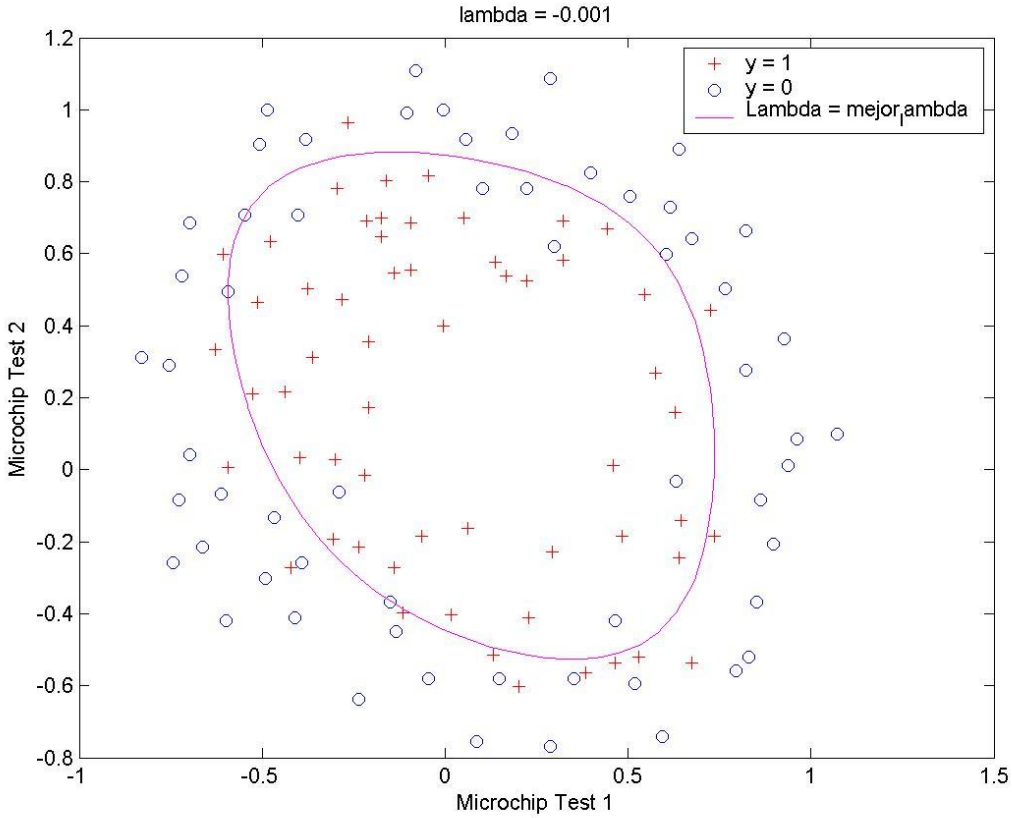
```
theta_mejor_lambda = minFunc(@CosteLogReg,theta_inicial,options,Xtr,ytr,mejor_lambda);
theta_lambda_0 = minFunc(@CosteLogReg,theta_inicial,options,Xtr,ytr,0);
```

Obtenidos los valores de theta, vamos a dibujar dos gráficas con las curvas de ambas regresiones. Ambas gráficas se han dibujado con todos los datos iniciales.

Lambda = 0



Lambda = -0.001



¿Qué podemos interpretar de estas dos gráficas?

Como podemos ver la regresión con valor de lambda = 0 se ajusta de una forma más compleja y precisa a los datos de entrenamiento, ya que dicho valor regulador, al ser nulo, produce sobreajuste al no regularizar la regresión.

Al contrario, si seleccionamos un valor de regularización, la regresión arroja una representación más lógica de la clasificación, eliminando así el sobreajuste producido por la expansión de los atributos.

Obtenidos los valores de theta para ambas regresiones vamos a comparar sus porcentajes de errores respecto de los datos de entrenamiento y test.

```
Errores con lambda=0
Tasa de errores con datos de entrenamiento: 7.368421
Tasa de errores con datos de test: 30.434783

Errores con lambda=-0.001000
Tasa de errores con datos de entrenamiento: 16.842105
Tasa de errores con datos de test: 21.739130
```

Como podemos ver la regresión con lambda=0 se ajusta mejor a los datos de entrenamiento, mientras que la regresión con el valor de lambda seleccionado se ajusta mejor con datos nuevos (datos de test), por lo que podemos afirmar que la regresión con valor regularizador (-0.001) es mejor al predecir mejor los datos nuevos.

4. Precisión/Recall

Para calcular los valores de precisión y recall hemos utilizado el modelo con lambda=-0.001.

Para calcular dichos valores debemos definir la matriz de confusión, compuesta por:

- TN : True negative
- FN : False negative
- FP : False positive
- TP : True positive

TP	FP
FN	TN

Para calcular dichos valores hemos definidos condiciones lógicas que deben cumplir los elementos del vector de salidas y el vector de salidas predichas.

```
%TN - True negative
tn = (sum(double((h==0)&(ycv==0))));
%FN - False negative
fn = (sum(double((h==0)&(ycv==1))));
%FP - False positive
fp = (sum(double((h==1)&(ycv==0))));
%TP - True positive
tp = (sum(double((h==1)&(ycv==1))));
```

La matriz de confusión obtenida tras la ejecución del programa es la siguiente-

```
matriz_confusion =

   11    2
    3    7
```

Obtenida la matriz, podemos obtener los valores de precisión y recall.

```
Precision = 0.846154
Recall = 0.785714
```

¿Qué podemos hacer para que el 95% de los chips aceptados sean buenos?

Para aumentar la precisión, es decir, aumentar el número de positivos que realmente son positivos, deberemos **incrementar el umbral** usado para la obtención de la predicción. El valor usado ahora mismo sería 0.5, si dicho valor aumentase podríamos hacer que la precisión aumente, acercándola así al valor pedido. En este caso tendríamos que elevar el umbral hasta que todas las muestras positivas sean reconocidas como positivas, es decir, obtener una matriz así:

```
matriz_confusion =

   14    2
    0    7
```

Si obtuviéramos otra matriz de confusión al modificar el umbral la precisión sería inferior al 95%.

```
matriz_confusion =

   13    2
    1    7
```

```
Precision = 0,928571
```