

# Programming Exercise 2: Logistic Regression

```
%Sigmoid Function
%Compute cost for logistic regression
%Gradient for logistic regression
%Predict function
%Compute cost for regularized LR
%Gradient for regularized LR
```

## Sigmoid Function

```
function h=sigmoid(z)
h=1./(1+exp(-z));
end
```

## Compute cost for logistic regression

```
function cost=Computecost(X,y,theta)
m=length(X);
part1=y'*log(sigmoid(X*theta));
part2=(1-y')*log(1-sigmoid(X*theta));
cost=(-1/m)*(part1+part2);
end
```

## Gradient for logistic regression

```
function grad=Gradient(X,y,theta)
m=length(X);
grad=(1/m)*X'*(sigmoid(X*theta)-y);
end
```

## Predict function

```
function p=Predict(X,theta)
p=round(sigmoid(X*theta));
end
```

## Compute cost for regularized LR

```
function costreg=ComputecostReg(X,y,theta,lambda)
m=length(X);
t=theta;
t(0)=0;
part1=y'*log(sigmoid(X*theta));
part2=(1-y')*log(1-sigmoid(X*theta));
costreg=(-1/m)*(part1+part2)+(lambda/(2*m))*(t'*t);
end
```

## Gradient for regularized LR

```
function gradreg=GradientReg(X,y,theta,lambda)
```

```
m=length(X);  
t=theta;  
gradreg=(1/m)*X'*(sigmoid(X*theta)-y)+(lambda/m)*t;  
end
```