**Project & Lab To Help Project & Remaining Labs:**

***Credit Card Fraud Detection***

Approximately 36% of organizations experienced economic crime. Therefore, there is definitely a need to solve the problem of credit card fraud detection. The task of fraud detection often boils down to outlier detection, in which a dataset is scanned through to find potential anomalies in the data. In the past, this was done by employees  which checked all transactions manually. With the rise of machine learning, artificial intelligence, deep learning and other relevant fields of information technology, it becomes feasible to automate this process and to save some of the intensive amount of labor that is put into detecting credit card fraud. In the following sections, my machine learning based Pythonic approach is explained.

##### We will use dataset from Kaggle (<https://www.kaggle.com/mlg-ulb/creditcardfraud/Data>). Kaggle has many other datasets and you use those for your project or labs.

**The datasets contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.**

**It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are ‘Time’ and ‘Amount’.**

**Feature ‘Time’ contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature ‘Amount’ is the transaction Amount. Feature ‘Class’ is the response variable and it takes value 1 in case of fraud and 0 otherwise.**

##### There are various solutions. I have picked the simplest one to get started. Later on we can add more complex versions that use multi-fold Cross-Validations, Grid Search and others.

##### We will use MLP and Logistic Regression (which we have not covered yet).

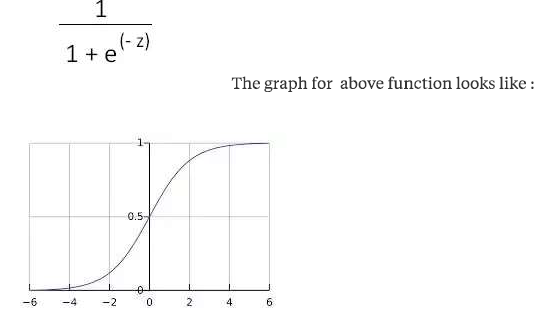
##### Loading and manipulating data:

##### Data file name is creditcard.csv. Show in Python.

##### Show code and explain in details using IPython.

**Logistic Regression**



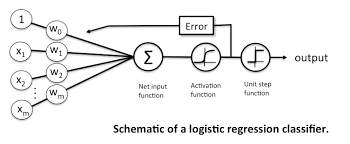


Since, **probability of any event to happen is [0,1] ( between 0 and 1, including both), this function definitely seems fit to be used as a probability function for logistic regression.**  
  
Now various question arise -  
1) **what is z** (input to sigmoid function) ?  
  
ans)           **z = transpose(theta) \* X**  
  
2) **what is X** here ?  
  
ans)      **X is the 'size of tumor**'. It is a tumor size that is not in inputs of our training set but we are trying to predict the probability for it to be malignant. Say X = 3.2



ML Versus Statistics (Some comparison):





Note that Logistic Regression is similar to Perceptron