Logistic regression

# Logistic regression theory

* Logistic regression is used for classification
* Can be used for diagnoses, debt default, etc.
* It is used typically to classify binary options where the dummy variables 1 and 0 are used to classify.
* A normal linear regression is not suitable for binary classification problem.
* A logistic regression curve can only go between 0 and 1.
* To do this, the sigmoid function is used because it can take in any value and output a value between 0 and 1 aways irrespective of the value that was inputted.
* The sigmoid function then allows setting a cut off point to use in classification of the data set.

## Model evaluation

* You can use a confusion matrix to evaluate the performance of a logistic model.
* A confusion matrix has 4 parts: True positive (TP), True Negative (TN), False Positive (FP – Type 1 error) and False Negative (FN -type 2 error).
* The interesting rates are: accuracy (TP+TN/total), misclassification or error rate (FP+FN/Total).

# Logistic regression – part 1 (Exploration)

* Kaggle is a website that hosts data sets and data science competitions.
* We will use titanic data set for this section
* You can check for missing data using a heatmap.
* If there are just a few missing data points, you can fill for them using knowledge from other columns, drop the entire column if its too many missing data points or simply explore the ratio of missing data in that column
* You can begin to explore the data starting from the target columns such as in this case, survived column before exploring the other relevant columns.
* You could use interactive plotly plots instead by importing cufflinks and using it offline with import cufflinks as cf and cf.go\_offline() respectively and then making the above plots with the .iplot keyword.

# Logistic regression – part 2 (Cleaning)

## Missing values replacement and dropping

* You can use imputation by filling the missing values with the average age of the passengers or the average age of their Pclass.
* You could also create a model to predict the age using all the other features and use the predictions to fill for the missing age values.
* In this instance, create a function to return age for missing age data using the Pclass, then, apply the function.
* You can drop the cabin column with df.drop(‘a’, axis=1, inplace = True) where df is data frame, a is column name and inplace ensures its done in the original data/
* For the embarked column, since its very few data that’s missing, we can drop the missing data with df.dropna(inplace=True)

## Handling categorical data by converting them to dummy variables

* Use the pd.get\_dummies() to convert categorical columns such as sex and embarked into dummies with a = pd.get\_dummies(a, drop\_first=True) where a is the column to be dummied and the drop\_first = True avoids multicollinearity (where one column is a perfect predictor of another e.g., any column that is true for male is automatically false for female and vice versa)
* Concatenate the new columns to the original data frame with pd.concat([a,b,c], axis = 1)
* After converting the categorical data to dummy variables, we can delete the original categorical data column as well as any columns that can’t be used for our purpose such as name etc with df.drop([a,b,c], axis = 1)
* Note: Pclass is actually a categorical data of 3 possible entries. It can be used as is or dummied. Explore the dummy option afterwards to see how models react to dummy variables vs categorical continuous data.

# Logistic regression – part 3(Training and predicting)

* Separate data to X and y and training and test set
* Use sklearn.model\_selection import train\_test\_split to split into training and test set.
* from sklearn.linear\_model import Logistic\_Regression and instantiate it e.g., lm = Logistic\_Regression()
* Train the model with lm.fit(X\_train, y\_train)
* Predict with lm.predict(X\_test)
* Evaluate the predictions with sklearn.metrics with either classification\_report or confusion\_matrix
* You can increase the max\_iter in Logistic\_Regression instance to avoid error of max iteration reached.