Intro to Linear Regression, Logistic Regression and K Nearest Neighbors

Linear Regression

sklearn.linear_model.LinearRegression

class sklearn.linear_model.**LinearRegression**(*, *fit_intercept=True, copy_X=True, n_jobs=None, positive=False*) [source]

Ordinary least squares Linear Regression.

LinearRegression fits a linear model with coefficients w = (w1, ..., wp) to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation.

- Continuous relationship between a dependent and independent variable
  - Predict the value of a continuous variable
- Multiple independent variables would be multilinear regression
- Example: population growth over time

$$Y = a + bX$$

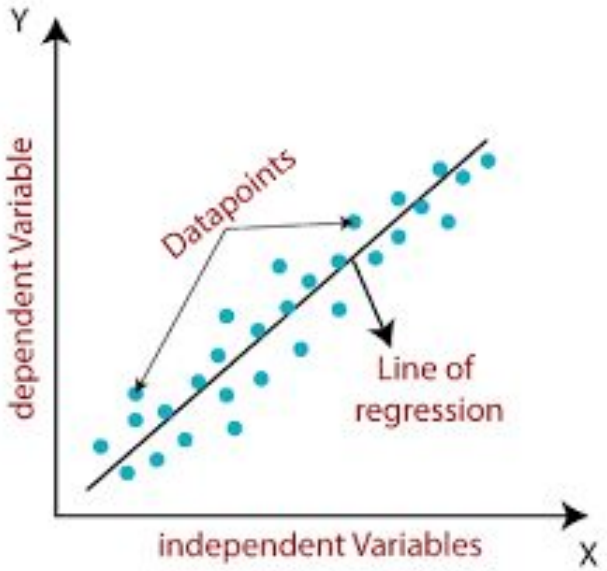
$$b = \frac{N\sum XY - (\sum X)(\sum Y)}{N\sum X^2 - (\sum X)^2} \qquad a = \frac{\sum Y - b\sum X}{N}$$

Where,

N = number of observations, or years

X = a year index (decade)

Y = population size for given census years



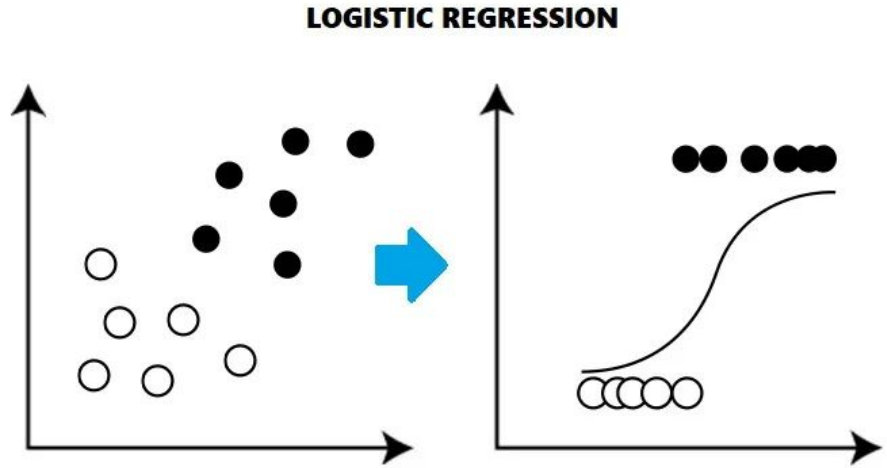
Logistic Regression

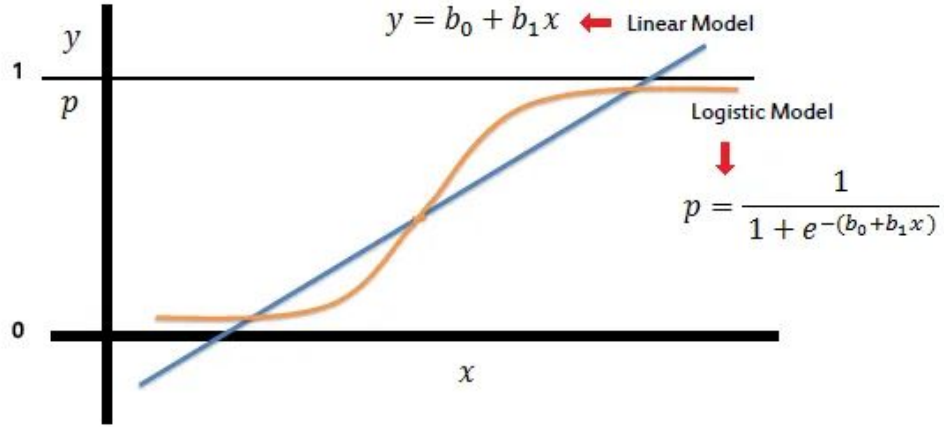
sklearn.linear_model.LogisticRegression

class sklearn.linear_model.**LogisticRegression**(*penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None*) [source]

Logistic Regression (aka logit, MaxEnt) classifier.

- Takes continuous data as input, and outputs a probability of classification
  - Classification can be binary (bacteria are or aren't motile) or multiclass (determine rock type from images and geochemical data)





https://medium.com/analytics-vidhya/understanding-logistic-regression-b3c672deac04

K Nearest Neighbors (kNN)

sklearn.neighbors.KNeighborsClassifier

class sklearn.neighbors.**KNeighborsClassifier**(*n_neighbors=5, *, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=None*) [source]

Classifier implementing the k-nearest neighbors vote.

sklearn.neighbors.KNeighborsRegressor

class sklearn.neighbors.**KNeighborsRegressor**(*n_neighbors=5, *, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=None*) [sour

Regression based on k-nearest neighbors.

The target is predicted by local interpolation of the targets associated of the nearest neighbors in the training set.

- Can be used for classification or regression (prediction of binary or continuous variables)
- Calculated based on the distance between dependent and independent variables
  - The nearest neighbors based on this distance calculation are then used for classification or regression

$$d(x, y_i) = \sqrt{\sum_{j=1, j \neq a}^{m} \left(x^{(j)} - y_i^{(j)}\right)^2}$$