Intro to Data Science
Classification and Clustering

1 Constructing a univariate Logistic Regression Curve

File to use: `Logistic regression`

The `Logistic regression-initial` worksheet shows the initial dataset and an empty curve, whereas `Logistic regression-final` shows the final optimal logistic regression curve along with unsubscribe probability for a given number of emails per week.

The dataset provides initial datapoints for the independent variable, Emails per Week, (values plotted on the X axis) and dependent variable, Unsubscribe (which is binary value, either 0% or 100%, plotted on the Y axis). To simulate more random data points, which are within the margin of either the 0% or 100%, check the Add Jitter checkbox.

Try setting various values for B0 and B1 parameters for the logistic regression curve equation (using only the arrow buttons to increase / decrease the values) and click on Show Model checkbox in order to view the regression curve for these different values. For each particular combination of values, you can see the Likelihood function computation for those parameter values.

The intention is to try and obtain a combination of values for the 2 parameters, so that the Likelihood is the largest possible. Try playing around with setting various values for these 2 parameters in order to get an intuition for what we are trying to achieve.

Finally, check on the Optimize checkbox. This produces a logistic regression curve whose B0 and B1 parameters will result in the LARGEST possible Likelihood value (594.4) for all possible combinations of values for these 2 parameters. This is effectively done using the Solver add on in Excel for optimization.
Next click on the + to view the hidden columns E and F. Column E contains the probability predicted for the corresponding X value (emails per week) using the logistic regression curve equation, while Column F contains the likelihood computation for that probability. Notice that for this optimal curve, the likelihood for the vast majority of predictions (whether they are close to 0 or close to 1) is nearly 1. This is what we have accomplished in the optimization process.

Finally, enter some sample values in the Email per week (1, 2, 3 …. 10), and check what is the predicted unsubscribe probability. You can see the boundary between the two binary possibilities (does not unsubscribe vs unsubscribe) is roughly around 8 (13.8%) and 9 (66.4%) emails.

A simple practical application of this univariate logistic regression model is to recommend to the marketing team to limit the number of promotion emails sent to customer to 8 or less.

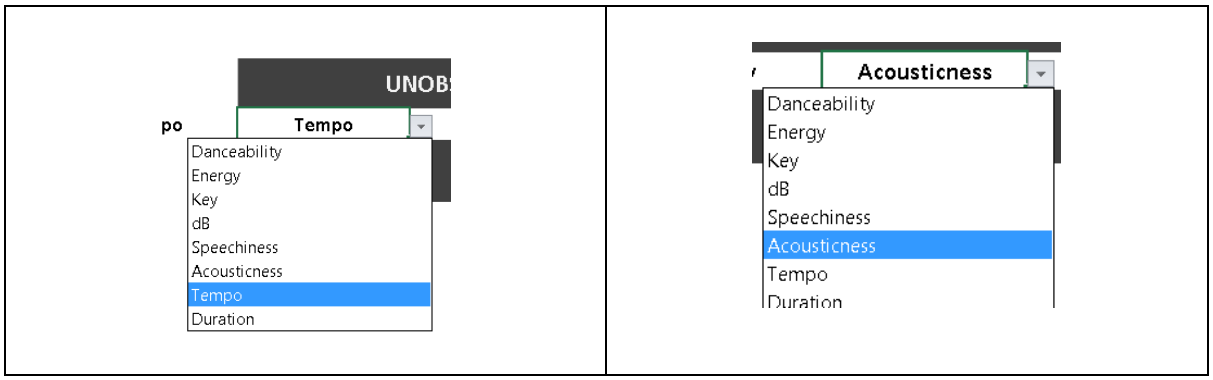
2 Implementing K-Nearest Neighbors (KNN)

File to use: `KNN`

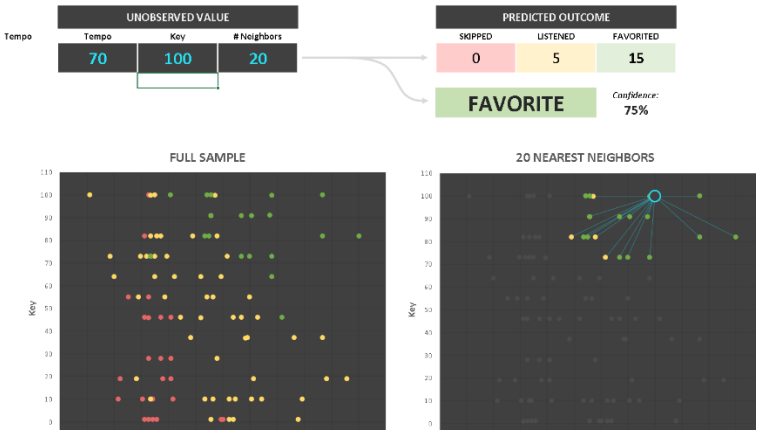
The `KNN-initial` worksheet shows the initial dataset without any unobserved value selected yet, whereas `KNN final` an unobserved value entered, along with a specified number of neighbors, and the predicted outcome for that unobserved value (which one of the 3 categories it is classified in) is provided.

If you expand columns M - U, you will notice that we do feature scaling (normalization) for all the independent variables of interest (Danceability, Energy, Key, dB, Speechiness, Acousticness, Tempo and Duration). This is to ensure that variables with large range (for e.g. duration) do not overwhelm the importance of variables with small range (for .eg. DB). Scaling ensuring values for all columns now fall within an identical range of 1 - 100.

You can select any of the two independent variables (from the total of 8 possible) to plot on a 2 dimensional graph for all the 3 possible outcome classes (Listened, Favorited, Skipped).



Next provide possible new values for a new data point in this graph for the 2 independent variables you selected as well as the value of K that you want to use, and you will be able to see the classification for your new data point into one of the 3 possible outcome classes (Listened, Favorited, Skipped), depending on which class has the most number of neighbors to the new data point.



3 Implementing K-Means Clustering

File to use: `K-Means`

The `K-means-initial` worksheet shows the initial dataset, whereas `K-means-final` shows the stage by stage operation of the K-Means algorithm through 4 rounds to reach the final grouping of clusters.

Make a copy of the `K-means-initial` worksheet. First, we need to identify how many clusters we wish to create, and we could use the WSS method to determine this, but for this simple example, we will just commence with 3 clusters.
Notice that we create 2 new columns containing the scaled values (normalization) for the two variables of interest that we are studying (Email Opens and Total Spent). This is to ensure the large range of the values of the Total Spent variable do not dominate the relevance of the smaller range of values of the Email Opens variable. Scaling ensuring values for both columns now fall within an identical range of 0 - 100.

Select any 3 rows whose values for both of the two variables being measured fall near at the start of the range, the middle of this range and at the end of this range. This would be a good starting point for the initial centroids to be used in this algorithm. An example might be:

1	2	9.38	$101.45	100.0	95.1
2	21	4.87	$62.01	42.3	48.8
3	36	1.64	$48.85	1.0	33.3

Once you have determined the locations of the 3 initial centroids, expand the hidden columns to see the step - by - step process of the K-means algorithm for 4 rounds to determining the final location of centroids and the cluster around those centroids. Technically, we should continue until the centroids no longer move between successive iterations, but to keep things simple in this example, we only proceed for 4 rounds.

Notice that for each iteration of this algorithm, the total distance for all the points from the centroid of their respective clusters keeps decreasing. This is conceptually correct, since when we eventually arrive at a fixed location for the centroids, the total distance should be the minimum.