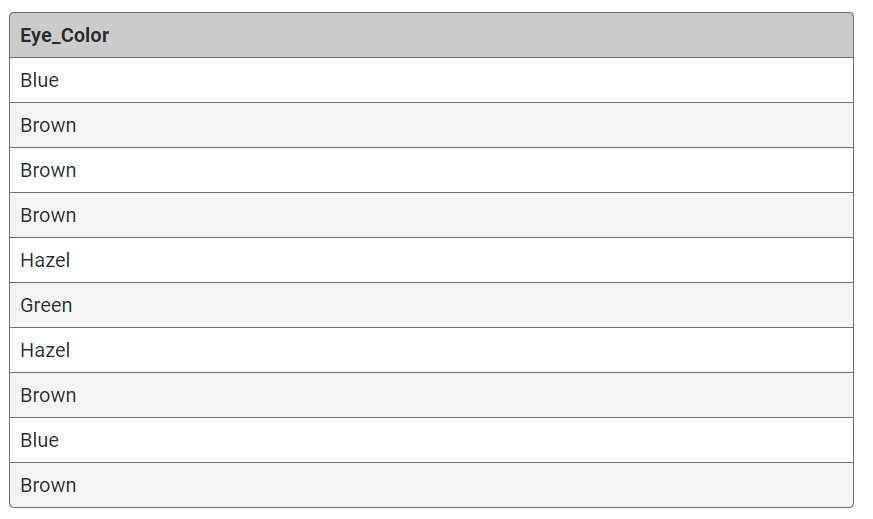
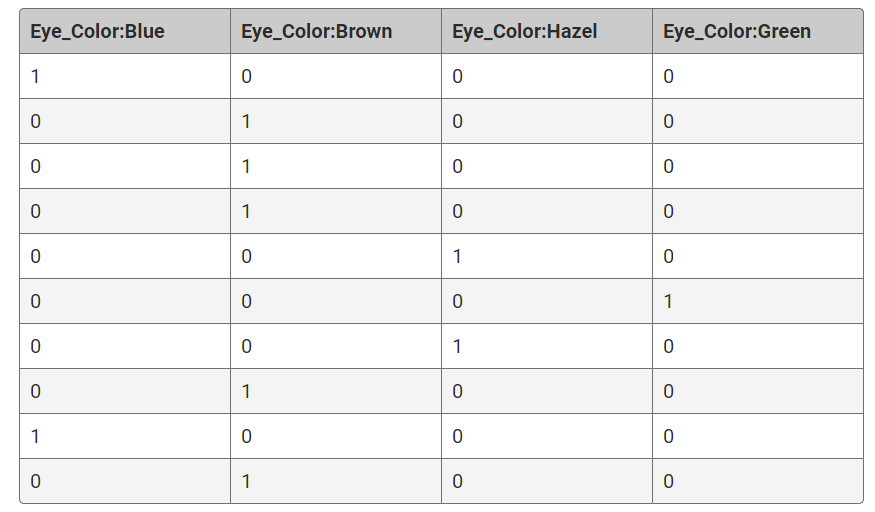
Unlike the statistical models such as multiple linear regression, or machine learning models such as random forest, neural networks cannot handle categorical variables in their raw form. Specifically, the perceptron neuron has no way to segment and keep track of all possible values in a categorical variable. So, what happens if our datasets contain categorical variables that are essential for identifying samples or groups of samples within the population data? Thankfully, there are straightforward solutions to grouping and encoding categorical variables without losing any information across neurons.

For a neural network to understand and evaluate a categorical variable, we must preprocess the values using a technique called **one-hot encoding**. One-hot encoding identifies all unique column values and splits the single categorical column into a series of columns, each containing information about a single unique categorical value. For example, let's imagine Beks received a dataset from a medical device company that contains clinical trial data. One of the variables in this dataset is a categorical variable that identifies the person's eye color:



If Beks were to apply the one-hot encoding to this categorical variable, it would break into four separate columns—one for each unique eye color (blue, brown, hazel, and green). Therefore, the newly encoded data columns would look as shown:



For each unique column value, every individual data point is evaluated, and if the categorical value matches, it is given the value of 1, otherwise it is 0. This binary encoding ensures that each neuron receives the same amount of information from the categorical variable. As a result, the neural network will interpret each value individually and provide each categorical value with an independent weight.

Although one-hot encoding is a very robust solution, it can be very memory-intensive. Therefore, categorical variables with a large number of unique values (or very large variables with only a few unique values) might become difficult to navigate or filter once encoded. To address this issue, we must reduce the number of unique values in the categorical variables. The process of reducing the number of unique categorical values in a dataset is known as **bucketing** or **binning**. Bucketing data typically follows one of two approaches:

1. Collapse all of the infrequent and rare categorical values into a single "other" category.
2. Create generalized categorical values and reassign all data points to the new corresponding values.

The first bucketing approach takes advantage of the fact that uncommon categories and "edge cases" are rarely statistically significant. Therefore, regression and classification models are unlikely to be able to use rare categorical values to produce robust models, and instead will ignore the rare events altogether and focus on more informative values.

The second bucketing approach collapses the number of unique categorical values and maintains relative order and magnitude so that the machine learning model can train on the categorical variable with minimal impact to performance. This approach is particularly useful when dealing with a categorical variable whose distribution of unique values is relatively even. Once we have bucketed our categorical variables, we can proceed to transform the categorical variable using one-hot encoding.