The Economic Impact of the World Cup

Jonah Witt, Michael Morales, and Taylor Moorman

IST 707

March 17, 2019

**Introduction**

[Insert Introduction]

**Analysis and Models**

*About the Data*

[Insert About the Data]

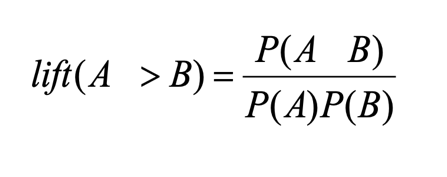
*Association Rule Analysis*

To gain information on the customers from the customer profile, it is important to look at the associations between different groups of customers. This is accomplished using association rule analysis. Association rule analysis is used for discovering relationships in large data sets. In this case, an association rule is a set of customer attributes that frequently occur together. These rules can consist of two or more attributes. Each rule will have two sides (left-hand side and right-hand side). The association rule shows the relationship between the attributes on the left-hand side and the attribute on the right-hand side. These association rules are measured based on the following:

*Support*: How often the rule occurs in the entire data set (percentage).

*Confidence*: How often the value in the right-hand side appears in the data that includes the value in the right-hand side (Value from 0 to 1).

*Lift*: The ratio of the confidence of the rule against the expected confidence of the rule. The formula for lift is shown below:



A rule must have a value greater than one to show that the rule has any meaning. Support,

confidence, and lift must all be taken into account when evaluation an association rule.

*k-Means Clustering*

K-Means clustering is a method used to group like data into “clusters” where the data points are most similar. The data are divided into a specified number (k) of clusters. There are k data points selected randomly (unless otherwise specified) as cluster centers. Afterwards, the remaining data points are grouped into the cluster where the point’s mean value is closest to the center of the cluster. If conducted properly, this will group the data points into clusters with similar characteristics.

*Cosine Similarity*

Cosine similarity is another method of comparing data. However, instead of clustering the data, the distance between each data point is measured. The formula for the cosine similarity between two points is shown below in Figure 3.



Figure 3: Cosine Similarity Formula

Once the distance is calculated between each point, once can see the point that is “most similar” to each data point.

*Decision Tree Analysis*

Decision Tree Analysis is a method of classifying data by splitting it into different label groups based on the values of various features. The data is broken off into “branches” from the main data set. These branches are determined by whether the data point meets certain criteria for a particular feature. For example, these decision nodes could represent whether the data record represents a male or female, a child or an adult, or any other split based off of a single feature. After the data is split, the data points reach the “terminal node” which classifies the data point based on the label in question. These trees can be tuned using various control measures. The two measures that were used to tune the trees in this analysis were minimum size of each split and the complexity parameter. The minimum size of each split indicates the minimum number of data points that must be in each “leaf” of the tree. The complexity parameter indicates how much the data point must improve the fit of the tree for the split to be included.

*Naïve Bayes*

Naïve Bayes is a method of classification based on Bayes Theorem. This theorem assumes that each predictor in a dataset is independent of one another. The model is considered “Naive” because of this assumption. Even if it is well known that the predictors are related to one another, the model still treats every predictor as entirely independent. These predictors are then used to predict the posterior probability. This can be shown in the equation below.

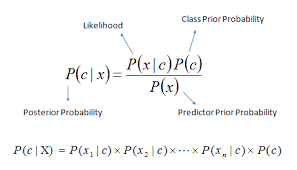


Figure 3: Posterior Probability Using Bayes Theorem

The assumption of independence allows the model to be run very quickly since the model does not have to calculate the probability for every single combination of predictors. However, this speed is a tradeoff for a more accurate model that takes the relationships of predictors into account.

*k-Nearest Neighbor*

The k-Nearest Neighbor algorithm is a method of supervised machine learning. This means that the label of the training data is known, and the model is looking to train the on the other features to classify the data on that label. In the k-Nearest Neighbor model, the Euclidean distance between two data points is used to classify a data point. Once the distances are calculated, the data point is classified by a vote of a specified number (k) of its nearest neighbors (The data points with the lowest Euclidean distance from the model). For example, if k was chosen to be one, the data point would be classified as whatever label the data point with the lowest Euclidean distance has. If k was chosen to be 5, the data point would be classified using the label of the majority of the 5 data points with the lowest Euclidean distance from the data point. This is shown below.

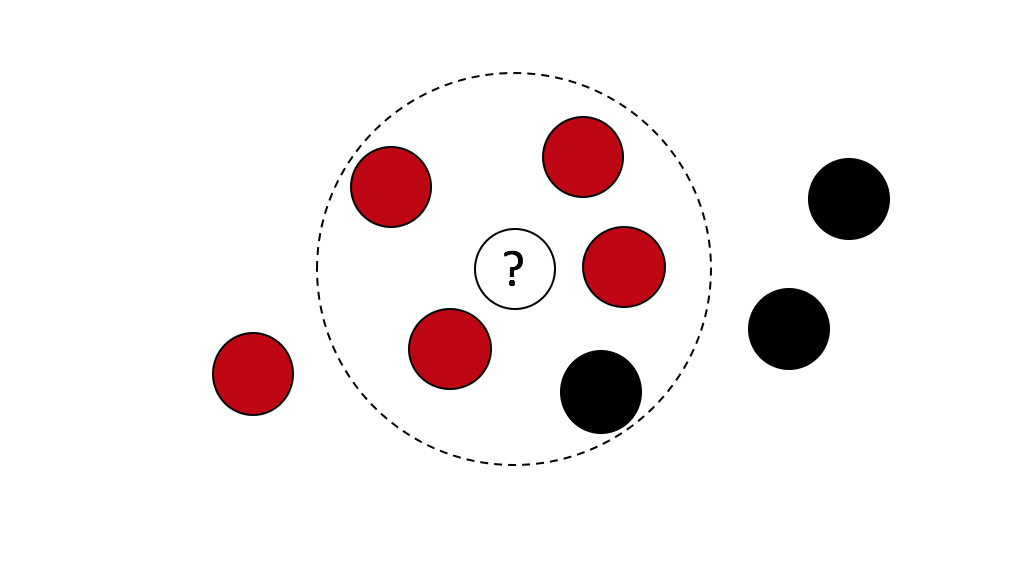


Figure : Example k-Nearest Neighbor

A larger k value can reduce the effects of noisy data. However, using too large of a k value can diminish the effects of smaller patterns in the data.

*Random Forest*

A random forest is a collection of randomly generated decision trees. However, these trees are not generated in the same manner as in Decision Tree analysis. When creating a single decision tree, the tree is created using all of the data and variables and then pruned to generate the best tree. A simple decision tree on a data set of Titanic Survivors is shown below.

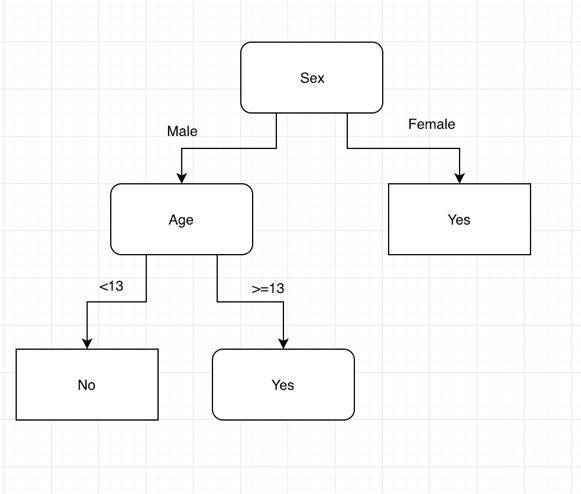


Figure : Decision Tree on Titanic Data

In decision tree analysis, each tree is created using a randomly selected group of data and variables. The output from these trees is combined to create the final output of the model. Each tree in the forest gets a “vote” to classify the data point. The label with the highest percentage of votes will classify the data point. This random method leads to a less biased result and typically performs better than a single decision tree. A simplified example from an article on Medium.com is shown below (Koehrsen 2017).



Figure : Simplified Random Forest Model

*Support Vector Machine*

The Support Vector Machine (SVM) is a supervised machine learning method used to classify binary data. This means that the SVM is used to split data into two groups with either “Label A” or “Label B”. This is achieved by creating a plane to separate the data. For data that is linearly separable, this plane is simply a line that maximizes the distance between both groups (the line that is evenly between the two groups).



Figure : Plane of Separation With Linearly Separable Data

While this seems simple for data that is linearly separable, this method becomes much more complicated when the data cannot be separated this way. In this case, the data will need to be transformed into a higher dimension. For example, consider the following example where the data appear to be separated by a circle.



Figure : Non-linearly Separable Data

In this example, the third-dimension *z* can be defined as z = x2 + y2 (This is the equation for a circle). A slice of this three-dimensional data can be shown below.



Figure : Z vs X Plot of Transformed Data

The linear plane can now be created to separate the data. When transformed back to the two-dimensional data, the plane of separation is shown below.



Figure : Plane of Separation Transformed Back to Original Dimensions

However, the method to define the third dimension is not often obvious. To accommodate this, different kernel functions can be used to train the model. These kernel functions can be linear, radial, polynomial, etc.

**Results**

*Association Rule Analysis*

[Insert AR results]

*k-Means Clustering*

[Insert k-Means results]

*Cosine Similarity*

[Insert CS results]

*Decision Tree Analysis*

[Insert DT results]

*Naïve Bayes*

[Insert NB results]

*k-Nearest Neighbor*

[Insert kNN results]

*Random Forest*

[Insert rf results]

*Support Vector Machine*

[Insert SVM results]

**Conclusions**

[Insert Conclusions]

**References**

Koehrsen, W., & Koehrsen, W. (2017, December 27). Random Forest Simple Explanation – Will Koehrsen – Medium. Retrieved March 5, 2019, from

https://medium.com/@williamkoehrsen/random-forest-simple-explanation-

377895a60d2d