**Red Wine Online Shop**

**Red Wine Online Shop** is e-commerce based in US which focuses on the quality issues of red wine. To make their job more efficient, **Red Wine Online Shop** wants to build a model that could predict the overall quality of the red wine from their ingredients. **Red Wine Online Shop** also wants to cluster customer’s intention from customer behavior in the website. Therefore, as a programmer, you are asked to help them build the application based on the existing dataset.

* + - 1. **Clustering (Self-Organizing Map)**

First, **Red Wine Online Shop** wants to **group the user** based on the **similarities** in user **behaviors**. To do that, you are going to use **Kohonen Self-Organizing Map** technique to **cluster the data**.

1. **Dataset Description**

**Content**

The given dataset contains **3,632 data** consisting of **behavior information** of all **Red Wine Online Shop** users in the web page.

**Feature Description**

The table below shows the feature descriptions in the dataset.

Table 1. Table of features descriptions for clustering

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Column** | **Description** | **Possible Value** |
| **Features** | Administrative | Administrative Value | Number |
| Administrative Duration | Duration in Administrative Page | Number |
| Informational | Informational Value | Number |
| Informational Duration | Duration in Informational Page | Number |
| Product Related | Product Related Value | Number |
| Product Related Duration | Duration in Product Related Page | 0 to 3 |
| Bounce Rates | Bounce Rates of a Web Page | Number |
| Exit Rates | Exit Rates of a Web Page | 0 to 3 |
| Page Values | Page Values of Each Web Page | Number |
| Special Day | Special Days Rate like Valentine etc. | String |
| Month | Month of the Year | String |
| Operating System | Operating System Used | 0 to 8 |
| Browser | Browser Used | 0 to 13 |
| Region | Region of the User | 1 to 9 |
| Traffic Type | Traffic Type in the Web Page | 1 to 20 |
| Visitor Type | Types of the Visitor | String |
| Weekend | Weekend or Not | True or False |
| Revenue | Revenue Will be Generated or Not | True or False |

1. **Feature Selection**

Instead of **using the actual value** for the clustering, you are asked to create **features derived** from the **actual data**. The features requested are:

Table 2. Required features and derivation formula

|  |  |
| --- | --- |
| **Feature** | **Derivation Formula** |
| Special Day Rate | if (Special Day is “HIGH”):  Special Day = 2  elif (Special Day is “NORMAL”):  Special Day = 1  elif (Special Day is “LOW”):  Special Day = 0 |
| Visitor Type | if (Visitor Type is “Returning\_Visitor”):  Visitor Type = 2  elif (Visitor Type is “New\_Visitor”):  Visitor Type = 1  elif (Visitor Type is “Other”):  Visitor Type = 0 |
| Weekend | if (Weekend is “TRUE”):  Weekend = 1  elif (Weekend is “FALSE”):  Weekend = 0 |
| Product Related Duration | Duration in Product Related Page |
| Exit Rates | Exit Rate of a Web Page |

1. **Feature Extraction**

After the five new features are extracted, you are asked to use **Principal Component Analysis** (**PCA**) to both clean the data and reduce the dimensionality even further.

The steps that you want to take are as follows:

1. **Select the features** as defined in the Feature Selection section
2. **Normalize** the data
3. Analyze the data with **Principal Component Analysis** to obtain the new components
4. Take the **highest 3 principal components** as the input of your neural network
5. **Architecture**

You are to **create your own architecture design** that will be **able to solve the given problem**. Consider the following when building your architecture:

* **Number of input nodes** required
* **Number of clusters**

These considerations will be **accounted for in the grading process**.

1. **Training**

The training procedure of the neural network are as follows:

* + - 1. **Epoch** forthetrainingsis **5000**
      2. **For each data** in the dataset, **find the winning node** by using **nearest distance**
      3. **Update the neighbor around** the winning node in a square pattern
      4. **Update the weight** of the network

1. **Visualization**

**After the training is complete**, use **matplotlib** to **visualize the clusters generated by the self-organizing map**.

The dataset is obtained from Kaggle (<https://www.kaggle.com/roshansharma/online-shoppers-intention>) by Roshan Sharma. The dataset has been heavily cleaned and modified for the purpose of this case.