|  |  |
| --- | --- |
| **Project Case** |  |
| COMP7117  Artificial Neural Network |
| **Computer Science** | **O192-COMP7117-AS01-00** |
| ***Valid on*** *Odd Semester Year 2018/2019* | **Revision 00** |

1. Seluruh kelompok tidak diperkenankan untuk:

*The whole group is not allowed to:*

* + - Melihat sebagian atau seluruh proyek kelompok lain,

*Seeing a part or the whole project from other groups*

* + - Menyadur sebagian maupun seluruh proyek dari buku,

*Adapted a part or the whole project from the book*

* + - Mendownload sebagian maupun seluruh proyek dari internet,

*Downloading a part or the whole project from the internet,*

* + - Mengerjakan soal yang tidak sesuai dengan tema yang ada di soal proyek,

*Working with another theme which is not in accordance with the existing theme in the matter of the project,*

* + - Melakukan tindakan kecurangan lainnya,

*Committing other dishonest actions,*

* + - Secara sengaja maupun tidak sengaja melakukan segala tindakan kelalaian yang menyebabkan hasil karyanya berhasil dicontek oleh orang lain / kelompok lain.

*Accidentally or intentionally conduct any failure action that cause the results of the project was copied by someone else / other groups.*

1. Jika kelompok terbukti melakukan tindakan seperti yang dijelaskan butir 1 di atas, maka **nilai kelompok** yang melakukan kecurangan (menyontek maupun dicontek) akan di – **NOL** – kan.

*If the group is proved to the actions described in point 1 above, the score of the group which committed dishonest acts (cheating or being cheated) will be “Zero”*

1. Perhatikan jadwal pengumpulan proyek, segala jenis pengumpulan proyek di luar jadwal tidak dilayani.

*Pay attention to the submission schedule for the project, all kinds of submission outside the project schedule will not be accepted*

1. Jangan lupa untuk melihat kriteria penilaian proyek yang ditempel di papan pengumuman, atau tanya asisten anda.

*Don’t forget to look at the project assessment criteria that posted on the announcement board, or ask your teaching assistant.*

1. Persentase penilaiaan untuk matakuliah ini adalah sebagai berikut:

*Marking percentage for this subject is described as follows:*

|  |  |  |
| --- | --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* | **UAP**  *Final Exam* |
| 40% | 60% | - |

1. Software yang digunakan pada matakuliah ini adalah sebagai berikut:

*Software will be used in this subject are described as follows:*

|  |
| --- |
| **Software**  *Software* |
| Microsoft Visual Studio Code  Python 3.6  Scipy  Scikit  TensorFlow 1.10 |

## Ekstensi file yang harus disertakan dalam pengumpulan tugas mandiri dan proyek untuk matakuliah ini adalah sebagai berikut:

*File extensions should be included in assignment and project collection for this subject are described as follows:*

|  |  |
| --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* |
| PY | PY |

## Soal

*Case*

**Bluejack AI Solutions**

You are working in **Bluejack AI Solutions**, an AI company that builds AI-based solutions to companies around the world. You are given a task to assist a hospital in a research that focuses on building a model that could predict the nutrition rating of a given food and clusters similar food from its nutritional content.

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

Diet of people around the world may be different in the amount of certain nutrition is contained in their food. They want to create a system that shows the **similarities of food** from **different countries** **based** on their **nutritional content**. To do that, you use **Kohonen Self-Organizing Map** technique to **cluster the data**.

* 1. **Dataset Description**

**Content**

The given dataset contains **1,000 data of food items** including their nutritional information. The data contains **200 food items** for **5 countries**: **Australia**, **Belgium**, **France**, **Switzerland**, and **United States**.

**Feature Description**

The table below shows the feature descriptions in the dataset.

Table 1. Table of Feature Descriptions for Clustering

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Column** | **Description** | **Possible Value** |
| **Features** | product\_name | The name of the product | String |
| brands | The brand that produces the product | String |
| countries\_en | The origin of the product | String |
| serving\_size | The manufacturer recommended serving size of the product | String |
| energy\_100g | The amount of energy contained in 100g serving | 0 to 15,000 |
| energy-from-fat\_100g | The amount of energy from fat contained in 100g serving | 0 to 4,000 |
| fat\_100g | The amount of fat contained in 100g serving | 0.0 to 100.0 |
| saturated-fat\_100g | The amount of saturated fat contained in 100g serving | 0.0 to 100.0 |
| monounsaturated-fat\_100g | The amount of monounsaturated fat contained in 100g serving | 0.0 to 100.0 |
| polyunsaturated-fat\_100g | The amount of polyunsaturated fat contained in 100g serving | 0.0 to 100.0 |
| omega-3-fat\_100g | The amount of omega 3 fat contained in 100g serving | 0.0 to 3.0 |
| trans-fat\_100g | The amount of trans fat contained in 100g serving | 0.0 to 500.0 |
| cholesterol\_100g | The amount of cholesterol contained in 100g serving | 0.0 to 2.0 |
| carbohydrates\_100g | The amount of carbohydrates contained in 100g serving | 0.0 to 100.0 |
| sugars\_100g | The amount of sugars contained in 100g serving | 0.0 to 100.0 |
| fiber\_100g | The amount of fiber contained in 100g serving | 0.0 to 100.0 |
| proteins\_100g | The amount of proteins contained in 100g serving | 0.0 to 100.0 |
| salt\_100g | The amount of salt contained in 100g serving | 0.0 to 100.0 |
| sodium\_100g | The amount of sodium contained in 100g serving | 0 to 50.0 |
| potassium\_100g | The amount of potassium contained in 100g serving | 0.0 to 80.0 |
| calcium\_100g | The amount of calcium contained in 100g serving | 0.0 to 15.0 |
| phosphorus\_100g | The amount of phosphorus contained in 100g serving | 0.0 to 2.0 |
| iron\_100g | The amount of iron contained in 100g serving | 0.0 to 25.0 |
| magnesium\_100g | The amount of magnesium contained in 100g serving | 0.0 to 0.5 |
| zinc\_100g | The amount of zinc contained in 100g serving | 0.0 to 0.1 |

* 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:

|  |  |
| --- | --- |
| **Feature** | **Derivation Formula** |
| Calorie Density | **calorie-density** = (energy + energy-from-fat) / 100 |
| Fat Ratio | **fat-ratio** = dividend / divisor, where   * dividend = (monounsaturated-fat + polyunsaturated-fat) * divisor = (saturated-fat + trans-fat)   if (divisor is 0):  divisor = 1 |
| Sugar | **sugars** |
| Protein | **proteins** |
| Salt | **salt** |

* 1. **Feature Extraction**

After the five 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 need 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 **three** **highest principal components** as the **input of your neural network**
   1. **Architecture**

You need 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**.

1. **Classification**

The hospital also wants to create a system which will classify whether the food is good for the body or not. To create such system, you are given a set of food items that have been labeled with Nutri-score grade; a **5-level grading system from A to E.**  **Score A** represents the **best** **food** for body, while **score** **E** represent the **least** **good** food for body.

* 1. **Dataset Description**

**Content**

The given dataset contains **10,000 data of food items** that are **already labeled with the Nutri-score system** devised by **Ministère des Solidarités et de la Santé (Ministry of Health of France)**. Each food item is also **accompanied by their nutritional information,** inwhich that nutritional information determines their Nutri-score grade. The grade itself refers to whether the food is good or bad for the body, with A being the best food item and E being the worst.

**Feature Description**

The table below shows the feature descriptions in the dataset.

Table 2. Table of Feature Descriptions for Classification

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Column** | **Description** | **Possible Value** |
| **Features** | product\_name | The name of the product | String |
| brands | The brand that produces the product | String |
| countries\_en | The origin of the product | String |
| serving\_size | The manufacturer recommended serving size of the product | String |
| energy\_100g | The amount of energy contained in 100g serving | 0 to 15,000 |
| energy-from-fat\_100g | The amount of energy from fat contained in 100g serving | 0 to 4,000 |
| fat\_100g | The amount of fat contained in 100g serving | 0.0 to 100.0 |
| saturated-fat\_100g | The amount of saturated fat contained in 100g serving | 0.0 to 100.0 |
| monounsaturated-fat\_100g | The amount of monounsaturated fat contained in 100g serving | 0.0 to 100.0 |
| polyunsaturated-fat\_100g | The amount of polyunsaturated fat contained in 100g serving | 0.0 to 100.0 |
| omega-3-fat\_100g | The amount of omega 3 fat contained in 100g serving | 0.0 to 3.0 |
| trans-fat\_100g | The amount of trans fat contained in 100g serving | 0.0 to 500.0 |
| cholesterol\_100g | The amount of cholesterol contained in 100g serving | 0.0 to 2.0 |
| carbohydrates\_100g | The amount of carbohydrates contained in 100g serving | 0.0 to 100.0 |
| sugars\_100g | The amount of sugars contained in 100g serving | 0.0 to 100.0 |
| fiber\_100g | The amount of fiber contained in 100g serving | 0.0 to 100.0 |
| proteins\_100g | The amount of proteins contained in 100g serving | 0.0 to 100.0 |
| salt\_100g | The amount of salt contained in 100g serving | 0.0 to 100.0 |
| sodium\_100g | The amount of sodium contained in 100g serving | 0 to 50.0 |
| potassium\_100g | The amount of potassium contained in 100g serving | 0.0 to 80.0 |
| calcium\_100g | The amount of calcium contained in 100g serving | 0.0 to 15.0 |
| phosphorus\_100g | The amount of phosphorus contained in 100g serving | 0.0 to 2.0 |
| iron\_100g | The amount of iron contained in 100g serving | 0.0 to 25.0 |
| magnesium\_100g | The amount of magnesium contained in 100g serving | 0.0 to 0.5 |
| zinc\_100g | The amount of zinc contained in 100g serving | 0.0 to 0.1 |
| **Output** | nutrition\_grade\_fr | The Nutri-score grade, which ranges from A for food with excellent nutritional content to E for food with worst nutritional content. | A, B, C, D, E |

* 1. **Feature Selection**

The Nutri-score system does not use all nutrition information in grading the quality of the food item. They tend to focus more on basic nutrition, such as: **the amount of energy contained, sugar, and fat**. However, the hospital wants you to expand the system to take additional nutritional information as consideration. This means that you need to add more features and cannot rely only on the base Nutri-score formula.

From the given dataset, here are the **features** that is going to be used by the model:

|  |
| --- |
| **Feature Name (Column)** |
| energy\_100g |
| energy-from-fat\_100g |
| fat\_100g |
| saturated-fat\_100g |
| monounsaturated-fat\_100g |
| polyunsaturated-fat\_100g |
| trans-fat\_100g |
| carbohydrates\_100g |
| sugars\_100g |
| proteins\_100g |
| salt\_100g |
| sodium\_100g |
| potassium\_100g |
| iron\_100g |

While the **output** of the system will be:

|  |
| --- |
| **Feature Name (Column)** |
| Nutri-score Grade (nutrition\_grade\_fr) |

* 1. **Feature Extraction**

Due to the **large number of features** that need to be considered in building the neural network, you want to **simplify the data** to make your network trains faster. While **reducing the complexity of the data is important**, **preserving the variance and relationship between the data is also important**. To solve those problems, your approach in **reducing the dimensionality** of the data is by using **Principal Component Analysis** technique.

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 5 principal components** as the **input of your neural network**
   1. **Architecture**

You need 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 output nodes** (classes) required
* **Whether hidden layer is required** **or not** (whether the case is a linearly separable case or not)

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

* 1. **Training**

The training of the neural network use **70% of the dataset that picked randomly**. The training is done with **gradient descent** as the optimization formula for **5,000 epochs**. In addition, during the training, **20% of the dataset** should be used as the **validation dataset**.

The training procedure are as follows:

1. **Initialization**

The initialization step needs to be run once before starting the training iteration:

1. Take the **output** of the **Principal Component Analysis** as the **features**
2. **Initialize** the **weights** and **biases** **randomly**

1. **Iteration**

For **5,000 epochs**, repeat the following:

1. **Calculate the error** by comparing the output of the neural network to the target in the dataset using **mean squared error** (**MSE**)
2. **Update** the **weights and biases** using **gradient descent optimization**
3. **For every 100 epochs**, **print** the **current error** and **epoch number** to the console
4. **After reaching the 500th epoch**, **calculate the validation error** by passing the validation dataset. After that, **record the validation error** and **save the model to file**
5. **For** **every 500 epochs**, **get the new validation error** by passing in the validation dataset. If the **validation error is lower** than the previous validation error, **save the model to file**. If the **validation error is higher**, **do not save the model**
   1. **Evaluation**

The neural network is to be **evaluated** based on the accuracy with **10% of the dataset** **after the training process** finished. The **accuracy** is calculated as follows:

|  |
| --- |
|  |

**Reference**

* The dataset is obtained from Open Food Facts, retrieved through Kaggle (https://www.kaggle.com/openfoodfacts/world-food-facts). The dataset has been heavily cleaned and modified for the purpose of this case.