|  |  |
| --- | --- |
| **Project Case** |  |
| COMP7117  Artificial Neural Network |
| **Computer Science** | **E212-COMP7117-RV02-00** |
| ***Valid on*** *Even Semester Year 2020/2021* | **Revision 00** |

1. Seluruh kelompok tidak diperkenankan untuk:

*The whole group is not allowed to:*

* + 1. Melihat sebagian atau seluruh proyek kelompok lain,

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

* + 1. Menyadur sebagian maupun seluruh proyek dari buku,

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

* + 1. Mendownload sebagian maupun seluruh proyek dari internet,

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

* + 1. 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,*

* + 1. Melakukan tindakan kecurangan lainnya,

*Committing other dishonest actions,*

* + 1. 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. Bila Anda tidak membaca peraturan ini, maka Anda dianggap telah membaca dan menyetujuinya

*If you have missed to read these regulations, so you are considered to have read and agreed on it*

1. Persentase penilaiaan untuk matakuliah ini adalah sebagai berikut:

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

|  |  |
| --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* |
| 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* |
| Visual Studio Code  Python 3.7  SciPy 1.5.0  SciKit 0.23.1  TensorFlow 1.15.0 |

## 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*

**Mush Farmoatic**

**Mush Farmoatic** is an e-commerce based in Europe which focuses on the quality and edibility of variety of mushrooms from their grown lab. To make their job more efficient, **Mush Farmoatic** wants to build a model that could predict the overall quality of the mushroom from their characteristic. **Mush Farmoatic** also wants to cluster similar edible mushroom from their characteristic. Therefore, as a programmer, you are asked to help them build the application based on the existing dataset.

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

First, **Mush Farmoatic** wants to **group the mushroom** based on the **similarities** in mushroom **physical feature**. 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 **8,124 data** consisting of **characteristic** of all **mushroom** found in wildernesss.

**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** | Cap Shape | Shape of mushroom cap | String |
| Cap Surface | Surface of mushroom cap | String |
| Cap Color | Color of mushroom cap | Number |
| Bruises | Possible bruises on mushroom | Boolean |
| Odor | Mushroom odor smell | String |
| Stalk Shape | Shape of mushroom stalk | String |
| Stalk Root | Type of stalk root | Number |
| Stalk Surface Above Ring | Type of stalk surface above ring | Number |
| Stalk Surface Below Ring | Type of stalk surface below ring | Number |
| Stalk Color Below Ring | Color of stalk below ring | Number |
| Stalk Color Above Ring | Color of stalk above ring | Number |
| Veil Type | Type of mushroom veil | String |
| Veil Color | Color of mushroom veil | Number |
| Ring Number | Number of stalk ring | 0 to 2 |
| Ring Type | Type of stalk ring | Number |
| Spore Print Color | Color of spore | Number |
| Population | Population of mushroom | String |
| Habitat | Habitat of mushroom | String |

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 . Required features and derivation formula

|  |  |
| --- | --- |
| **Feature** | **Derivation Formula** |
| Bruises | Possible bruises of mushroom |
| Odor | if (Odor Type is “a”):  Odor Type = 1  elif (Odor Type is “l”):  Odor Type = 2  elif (Odor Type is “c”):  Odor Type = 3  elif (Odor Type is “y”):  Odor Type = 4  elif (Odor Type is “f”):  Odor Type = 5  elif (Odor Type is “m”):  Odor Type = 6  elif (Odor Type is “n”):  Odor Type = 7  elif (Odor Type is “p”):  Odor Type = 8  elif (Odor Type is “s”):  Odor Type = 9 |
| Stalk Shape | if (Stalk Shape is “e”):  Stalk Shape = 1  elif (Stalk Shape is “t”):  Stalk Shape = 2 |
| Veil Type | if (Veil Type is “p”):  Veil Type = 1  elif (Veil Type is “u”):  Veil Type = 2 |
| Spore Print Color | Color of spore |

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 **2500**
      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**

In addition to **cluster similar edible mushroom traits**, **Mush Farmoatic** also wants you to create an application that can help them **classify** the **status** of the mushroom they sell. In order to do that, you will be given a dataset labeled with **overall status of the mushroom** which depends on various traits of the mushroom.

* 1. **Dataset Description**

**Content**

The given dataset contains **8,124 data of the mushroom** which are labeled with **overall traits of the mushroom** organized by **UCI Machine Learning**. The overall edibility is based on the final result of the mushroom, with either “**Edible**” or “**Poisonous**”

**Feature Description**

The table below shows the feature descriptions in the dataset.

Table . Table of feature descriptions for classification

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Column** | **Description** | **Possible Value** |
| **Features** | Cap Shape | Shape of mushroom cap | String |
| Cap Surface | Surface of mushroom cap | String |
| Cap Color | Color of mushroom cap | Number |
| Bruises | Possible bruises on mushroom | Boolean |
| Odor | Mushroom odor smell | String |
| Stalk Shape | Shape of mushroom stalk | String |
| Stalk Root | Type of stalk root | Number |
| Stalk Surface Above Ring | Type of stalk surface above ring | Number |
| Stalk Surface Below Ring | Type of stalk surface below ring | Number |
| Stalk Color Below Ring | Color of stalk below ring | Number |
| Stalk Color Above Ring | Color of stalk above ring | Number |
| Veil Type | Type of mushroom veil | String |
| Veil Color | Color of mushroom veil | Number |
| Ring Number | Number of stalk ring | 0 to 2 |
| Ring Type | Type of stalk ring | Number |
| Spore Print Color | Color of spore | Number |
| Population | Population of mushroom | String |
| Habitat | Habitat of mushroom | String |

* 1. **Feature Selection**

To **classify** the **overall rating** of the hospital, **Red Wine Online Shop** does not want to use all features. They will select **only a few features** for you to build the application. Instead of using the **actual value**, you are tasked to create **features derived** from the actual data. The features requested are:

Table . Required features and derivation formula

|  |  |
| --- | --- |
| **Feature** | **Derivation Formula** |
| Cap Shape | if (Cap Shape is “b”):  Odor Type = 1  elif (Odor Type is “c”):  Odor Type = 2  elif (Odor Type is “x”):  Odor Type = 3  elif (Odor Type is “f”):  Odor Type = 4  elif (Odor Type is “k”):  Odor Type = 5  elif (Odor Type is “s”):  Odor Type = 6 |
| Cap Color | Color of mushroom cap |
| Odor | if (Odor Type is “a”):  Odor Type = 1  elif (Odor Type is “l”):  Odor Type = 2  elif (Odor Type is “c”):  Odor Type = 3  elif (Odor Type is “y”):  Odor Type = 4  elif (Odor Type is “f”):  Odor Type = 5  elif (Odor Type is “m”):  Odor Type = 6  elif (Odor Type is “n”):  Odor Type = 7  elif (Odor Type is “p”):  Odor Type = 8  elif (Odor Type is “s”):  Odor Type = 9 |
| Stalk Color Above Ring | Color of stalk above ring region |
| Stalk Color Below Ring | Color of stalk below ring region |
| Veil Color | Color of mushroom veil |
| Ring Number | Number of rings occurred in mushroom |
| Habitat | if (Habitat is “g”):  Habitat = 1  elif (Habitat is “l”):  Habitat = 2  elif (Habitat is “m”):  Habitat = 3  elif (Habitat is “p”):  Habitat = 4  elif (Habitat is “u”):  Habitat = 5  elif (Habitat is “w”):  Habitat = 6  elif (Habitat is “d”):  Habitat = 7 |

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

Table . Required system output

|  |
| --- |
| **Label Name** |
| Mushroom Status (Consumability) |

* 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 (PCA)** 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 4 principal components** as the **input of your neural network**
   1. **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 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 is done with **70% of the dataset picked randomly**. The training is done with **gradient descent** as the optimization formula for **2,500 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**
3. **Iteration**

For **2,500 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 25 epochs**, **print** the **current error** and **epoch number** to the console
4. **After reaching the 125th 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 125 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 Kaggle (https://www.kaggle.com/uciml/mushroom-classification) by UCI Machine Learning. The dataset has been heavily cleaned and modified for the purpose of this case.